NNT : 2019SACLE027

THÈSE DE DOCTORAT

de

L'Université Paris-Saclay

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : Université d'Evry-Val d'Essonne

Laboratoire d'accueil : Laboratoire de mathématiques et modélisation d'Évry, UMR 8071 CNRS-INRA

Spécialité de doctorat : Mathématiques appliquées

Babacar Diallo

X-Valuation Adjustments Computations by Nested Simulation on Graphics Processing Units.

Date de soutenance : 9 Décembre 2019

Après avis des rapporteurs : Aurélien Alfonsi (ENPC)
Jérôme Lelong (Grenoble INP - Ensimag)

Jury de soutenance :	Lokmane Abbas-Turki	(Sorbonne Université) Co-encadrant de thèse
	Aurélien Alfonsi	(ENPC) Rapporteur
	Stéphane Crépey	(Université d'Evry Val d'Essone) Directeur de thèse
	Agathe Guilloux	(Université d'Evry Val d'Essone) Présidente du jury
	Christophe Michel	(CACIB) Invité
	Moez Mrad	(CACIB) Invité
	Gilles Pagès	(Sorbonne Université) Examinateur

.

Thèse réalisée au Laboratoire de Mathématiques et Modélisation d'Évry, 23 bvd de France, 91037 Évry Cedex, FRANCE.

Et au Laboratoire de Probabilités, Statistique et Modélisation, 4 place Jussieu, 75252 PARIS, France.

Ainsi qu'à Crédit Agricole CIB, 12 Place des Etats-Unis, 92547 Montrouge, France.

Directeur de thèse : Stéphane Crépey, Professeur des Universités, Université d'Evry Val d'Essone.

Co-encadrant de thèse : Lokmane Abbas-Turki, Maitre de Conférence, Sorbonne Université.

Encadrement Entreprise : - Christophe Michel, Responsable recherche quantitative, CACIB.
- Moez Mrad, Responsable XVA desk, CACIB.

Financement Convention CIFRE entre la banque CACIB, le laboratoire LaMME, le laboratoire LPSM et l'Association Nationale de la Recherche et de la Technologie (ANRT).

Remerciements

Tout d'abord je rends grâce à Dieu de m'avoir donné le courage et la force de m'engager et d'arriver au bout de ce projet. Je voudrais profiter de ces quelques lignes pour remercier les personnes qui m'ont apportées leur soutien durant ces trois années de recherche.

Je voudrais tout d'abord remercier Lokmane Abbas-Turki de m'avoir fait confiance depuis mon stage de fin d'études. Je le remercie d'avoir initié cette thèse, mais surtout pour son encadrement et aussi d'avoir été à l'origine du dernier chapitre de ce travail qui m'a beaucoup appris. Je voudrais remercier Stéphane Crépey d'avoir accepté de diriger cette thèse. Je voudrais aussi le remercier pour la qualité de son encadrement et nos différentes discussions très enrichissantes, particulièrement sur le sujet des XVAs. Je remercie aussi Christophe Michel d'avoir rendu possible cette thèse CIFRE mais aussi d'avoir accepté de m'encadrer au sein de Crédit Agricole CIB. Nos réunions m'ont permis de comprendre les problèmatiques posés par les XVAs au sein de la banque. Je vous remercie tous les trois de m'avoir soutenu durant les périodes difficiles.

Je voudrai aussi remercier Moez Mrad qui m'a encadré au sein de CACIB et qui m'a appris beaucoup de choses sur les problèmatiques de calcul de risque de contrepartie. Son encadrement a été essentiel à la réussite de cette thèse. L'adaptation de la régression "proxy on indicator", pour le calcul de la FVA avec collatéral, dans l'introduction a été inspiré par lui.

Je souhaite également exprimer toute ma gratitude envers Aurélien Alfonsi et Jérôme Lelong pour le temps qu'ils m'ont accordé en tant que rapporteurs de ce manuscrit. Je tiens également à exprimer toute ma reconnaissance envers Agathe Guilloux, Moez Mrad et Gilles Pagès d'avoir accepté de faire partie de mon jury de thèse.

Je tiens ensuite à remercier les personnes avec lesquels j'ai collaboré : David Barrera, Gersende Fort, Emmanuel Gobet, Moez Mrad et Uladzislau Stazhynski. Je remercie aussi les membres du LaMME, particulièrement Yannick, Wissal, Maouloud, Valérie, Bouazza, Abass, ainsi que du LPSM, particulièrement Thibaut, Nicolas (les deux), Rancy, Sarah, Eric, Florence tous les doctorants, et enfin l'équipe de recherche quantitative de CACIB.

Pour finir je voudrais remercier les personnes partageant ma vie personnelle et qui m'ont soutenu. Je remercie Badiane Mbossé et toute sa famille, qui ont été d'un soutien essentiel depuis ma venue en France. Je remercie aussi mes amis et ma famille. Je termine en remerciant mes parents Ousmane Diallo, Alimatou Ba, Aminata Ba et ma femme Maimouna Faye.

Table des matières

Chapitre 1
Introduction

En 2007 une crise financière est survenue dans le monde déclenchée par la chute des prêts hypothécaires aux Etats Unis, appelée crise des «subprimes». Cet événement majeur a créé une instabilitée sans précédente dans les marchés financiers et a remis en cause tout le systéme financier. En effet, certaines institutions financières qui étaient considérées comme «trop grand pour faire faillite» ont subi des pertes financières tellement importantes qu'elles ont dû déclarer faillite. Cela a causé des pertes financières importantes pour leurs clients et par un effet domino atteint tout le système financier mondial. La faillite de la banque Lehman Brothers le 15 septembre 2008 est sans doute l'exemple qui illustre le mieux cette crise et ses conséquences.

Une analyse des causes de cette crise a permis de relever d'importantes failles du système financier. L'une d'entre elles est que le risque de défaut des contreparties n'était pas pris en compte dans la valorisation des produits dérivés dans les transactions de gré à gré (OTC). L'exposition des banques à ce risque de défaut est appelée risque de contrepartie. De plus les transactions OTC avaient une part de marché très importante dans la finance (voir Figure 1.1). Par ailleurs, la survenue d'évènements extrêmes tels qu'une crise financière n'était pas en adéquation avec les méthodologies de mesure de risque et de calcul de fonds propres utilisés. En effet, les mesures de risques n'étaient pas fiables en période de stress des marchés. La complexité des modèles mathématiques utilisés dans la valorisation des produits dérivés a elle aussi était pointée du doigt.

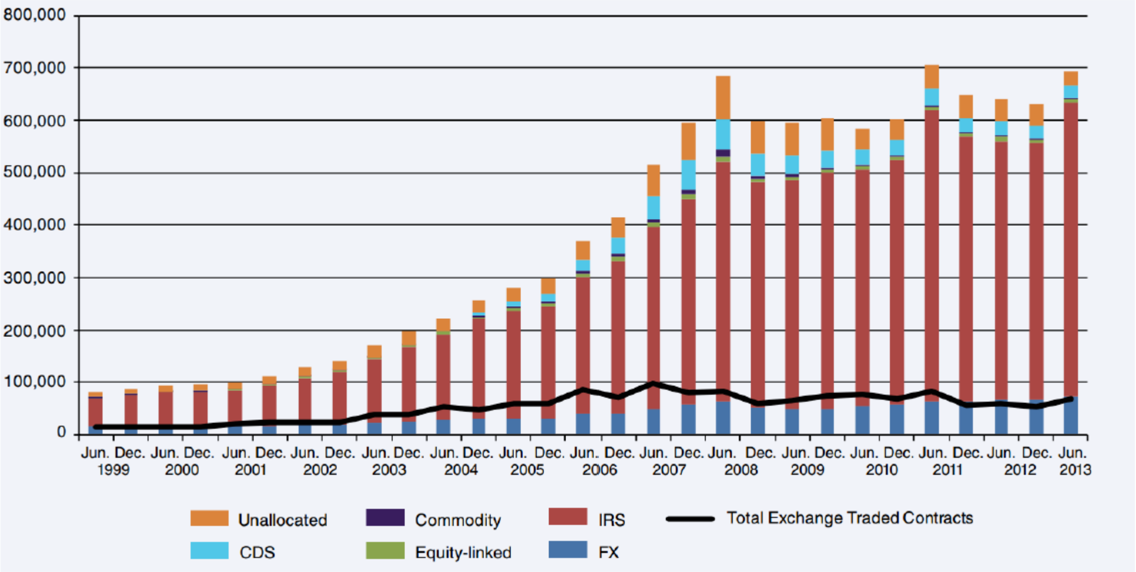


Figure 1.1 – Evolution marché OTC en milliard de dollars US (source BIS https ://www.bis.org)

D'autre part la crise financière a entrainé un décalage des taux Libor[1], jusque

[1] Taux interbancaire

là considérés comme sans risque, par rapport au taux OIS[2]. Ce décalage est dû essentiellement à une crise de confiance sur le marché interbancaire. La figure 1.2 montre l'impact de la crise sur le spread Libor/OIS.

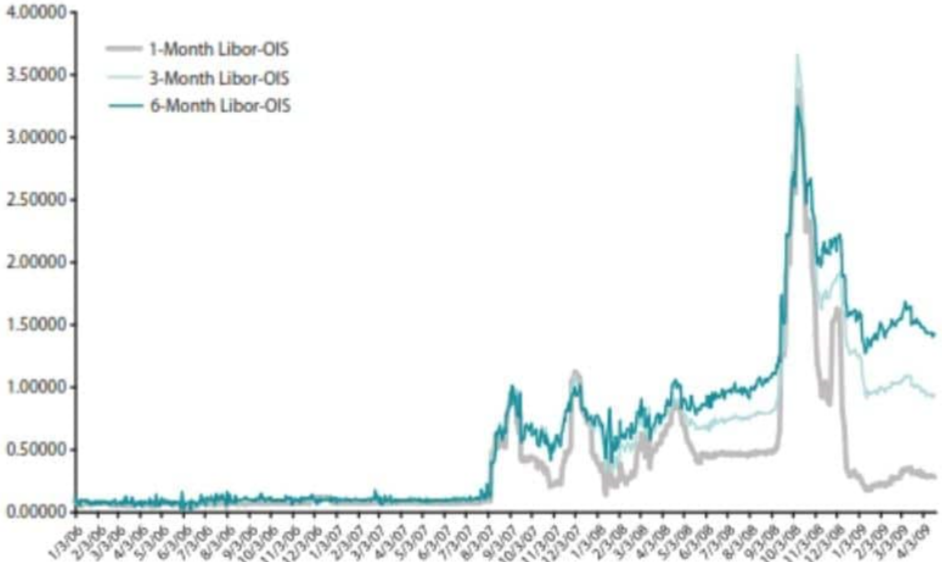


FIGURE 1.2 – Cours du spread Libor/OIS (Source : Federal Reserve Bank of St. Louis)

Dès lors les régulateurs du systéme financier ont mis en place un nouveau cadre réglementaire[3] afin d'éliminer les failles du système et de réduire les risques de crise financière dans le futur. Les principales propositions sont les suivantes :

1 - La centralisation des transactions OTC au sein de chambres de compensation (CCP[4]). Ces instituts permettent de réduire considérablement le risque de contrepartie. En effet les chambres de compensation se placent comme interface au milieu de toutes les transactions (voir Figure 1.3). Cela permet non seulement d'organiser le marché OTC, mais aussi de réduire les effets de contagion de défaut entre les banques participantes (clearing members).

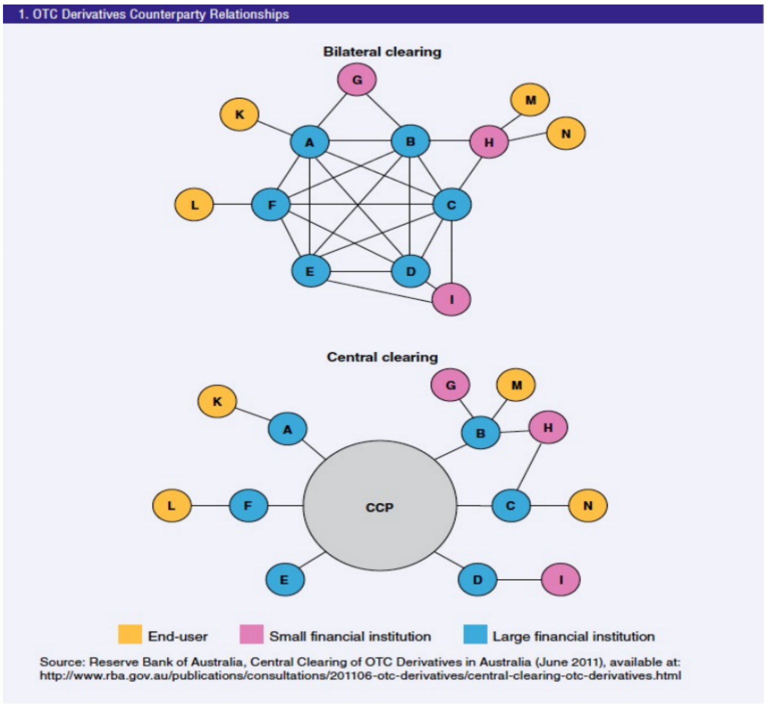
2 - La mise en place d'un système d'échange de collatéral et d'appels de marges entre les contreparties des transactions OTC. Ces échanges de collatéral permettent de réduire la perte potentielle dû au risque de contrepartie dans le cadre des transactions OTC.

Les banques ont réagi à ces évolutions réglementaires par l'introdution d'ajustements sur la valorisation des produits dérivés. Ces ajustements permettent de prendre en compte non seulement le coût du risque de contrepartie mais aussi le coût de financement du collatéral et le coût du capital. L'ensemble de ces ajustements est appelé XVA, où VA renvoie à ajustement de valorisation, et X peut être : C pour le crédit, D pour le débit, F pour le financement de la marge dite de variation, M pour le financement de la marge dite initiale, et K pour le capital.

[2] Overnight Indexed Swap

[3] FRTB, Bâle II, Bâle III, ISDA, ...

[4] Central Counterparty


1. OTC Derivatives Counterparty Relationships
Bilateral clearing
G
K
M
A
B
H
N
L
F
C
E
D
I
Central clearing
G
M
K
A
B
H
L
F
CCP
C
N
E
D
I
🟨 End-user
🟪 Small financial institution
🟦 Large financial institution

Source: Reserve Bank of Australia, Central Clearing of OTC Derivatives in Australia (June 2011), available at: http://www.rba.gov.au/publications/consultations/201106-otc-derivatives/central-clearing-otc-derivatives.html

FIGURE 1.3 – Transaction bilateral vs transaction en CPP (Source : Reserve bank of Australia, 2011).

Cette thése traite des problèmatiques de calcul posées par les XVAs. Ces indicateurs ont un rôle clé pour la gestion du risque de contrepartie. Ils constituent un nouveau paradigme dans la finance de marché tant par leur définition, leur modélisation que leur calcul numérique. Dans la section 1.1 nous introduisons et modélisons ces indicateurs dans un cadre général. Dans la section 1.2 nous présentons les méthodes de calculs de XVAs basées sur la régression. Dans la section 1.3 nous présenterons les méthodes de calcul par Monte Carlo emboîté explorées dans cette thése.

1.1 Introduction sur les XVAs

Nous considérons la base de pricing stochastique $(\Omega, \mathbb{G}, \mathbb{P})$ avec une filtration $\mathbb{G}$ et une mesure de probabilité risque neutre $\mathbb{P}$, tous les processus sont $\mathbb{G}$ adaptés. Nous notons r le taux sans risque et $\beta = e^{-\int_0^{\cdot} r_s ds}$ le facteur d'actualisation correspondant. L'espérance conditionnelle est notée $\mathbb{E}_t$.

Une banque B rentre en transaction avec un client C. Cette transaction est une portefeuille de produits dérivés. Nous notons par V_t la valeur du portefeuille (hors risque de contrepartie) à l'instant t de maturité finale T. B et C ont des temps de défaut τ_B et τ_C. Nous notons VM, PIM, RIM la marge de variation échangée entre la banque et le client, la marge initiale postée par la banque et la marge initiale reçue par la banque. Nous supposons que la VM est positive si elle est reçue par la banque et négative si elle est postée par la banque.

1.1.1 Credit Valuation Adjustment (CVA)

La CVA est la perte attendue de B sur le portefeuille en cas de défaut de C. Pour gérer ce risque la banque va ajuster le prix de la transaction en facturant au client

la valeur de la CVA. Il est de ce fait important de bien modéliser la CVA.
Cette CVA se définit comme suit, à toute date $t \geqslant 0$:

$$\mathrm{CVA}_t = (1-R)\mathbb{E}_t\left[\mathbf{1}_{t<\tau_C<T}\beta_t^{-1}\beta_{\tau_C}(V_{\tau_C} - \mathrm{VM}_{\tau_C} - \mathrm{RIM}_{\tau_C})^+\right], \tag{1.1.1}$$

où R est le taux de recouvrement du client, et on note $x^+ = \max(0, x)$.
La présence de R dans la formulation de la CVA s'explique par le fait que si le client fait défaut la banque recevra un recouvrement d'un taux R de son exposition à l'instant de défaut du client. Cela explique que la CVA n'estime que l'exposition sur la partie non recouvré $1-R$. Dans les calculs de CVA le taux de recouvrement est en pratique pris égal à 40%.

Cette formulation de la CVA se simplifie si le taux risque neutre r est déterministe et que le défaut τ_B est indépendant de l'exposition du portefeuille. Dans ce cadre la CVA peut s'écrire à l'instant $t=0$ comme suit :

$$\mathrm{CVA}_0 = (1-R)\int_0^T \beta_t \mathrm{EPE}(t)\mathbb{P}(\tau_C \in dt), \tag{1.1.2}$$

où $\mathrm{EPE}(t) = \mathbb{E}(V_t - \mathrm{VM_t} - \mathrm{RIM_t})^+$.
Mais sur certains portefeuilles il existe une corrélation adverse entre l'exposition et le défaut, appelée wrong way risk (WWR).

1.1.2 Debit Valuation Adjustment (DVA)

La DVA représente la CVA du point de vue du client. Elle correspond à la perte potentielle à laquelle la banque expose celui ci si la banque elle même fait défaut. Cette perte représente un gain pour la banque mais un gain purement comptable qui n'est donc pas pris en compte dans le pricing.

1.1.3 Funding Valuation Adjustment (FVA)

La FVA correspond aux coûts des emprunts de la banque pour financer sa marge de variation. La banque facture au client sa FVA.
La FVA est modélisée comme suit :

$$\mathrm{FVA}_t = \mathbb{E}_t\int_t^{T\wedge\tau_C} \beta_t^{-1}\beta_s\lambda_s(V_s - \mathrm{VM_s} - \mathrm{CVA_s} - \mathrm{FVA_s})^+\mathrm{ds}, \tag{1.1.3}$$

où λ_s est le spread de financement de la banque.
La prise en compte de la FVA a été remise en question par Hull et White [77]. Cependant la plupart des praticiens s'accordent à dire que la FVA doit être prise en compte dans la valorisation.

1.1.4 Margin Valuation Adjusmtent (MVA)

La MVA est le coût de financement de la marge initiale (IM) posté par la banque pour faire face à la perte potentielle sur le portefeuille durant la période de liquidation après un défaut de la banque.
Elle est modélisée comme suit :

$$\mathrm{MVA}_t = \mathbb{E}_t\int_t^{T\wedge\tau_C} \beta_t^{-1}\beta_s\lambda_s\mathrm{PIM}_s ds, \tag{1.1.4}$$

où PIM est la marge initiale postée par la banque. L'ISDA a mis en place un modèle appelé Standard Initial Margin Method (SIMM) qui permet de standardiser le calcul des marges initiales dans le cadre des transactions bilatérales.

1.1.5 Capital Valuation Adjustment (KVA)

La KVA est le coût du capital mis à risque par les actionnaires de la banque pour faire face à des pertes exceptionnelles.
Une première étape est de modéliser le capital économique (EC[5]). Ce capital est défini comme une moyenne des pires scénarios de perte sur une année. On le modélise par un expected shortfall ($\mathbb{ES}$) qui s'écrit comme suit :

$$\mathrm{EC}_t = \mathbb{ES}_t^a\Big(\int_t^{t+1} \beta_t^{-1}\beta_s dL_s\Big), \tag{1.1.5}$$

où a est le niveau de quantile de l'expected shortfall et L est la perte de trading de la banque. L'expected shortfall au seuil a est défini par la perte de trading :

$$\mathbb{ES}_t^a(X) = \frac{1}{(1-a)}\int_a^1 \mathbb{V}a\mathbb{R}_t^\alpha(X)d\alpha$$

où $\mathbb{V}a\mathbb{R}_t^\alpha(X) = \inf\{x \in \mathbb{R} : \mathbb{P}(X \leq x \mid G_t) \geq \alpha\}$. Partant de cette définition de l'EC il est dès lors possible de définir la KVA, pour un taux de rendement noté h, par

$$\mathrm{KVA}_t = h\mathbb{E}_t\left(\int_t^T e^{\int_t^s (r_u+h)du}\mathrm{EC}_s ds\right). \tag{1.1.6}$$

1.2 Calcul des XVAs par méthode de régression

Après avoir défini les XVAs nous allons nous intéresser à leur calcul.
La méthode traditionnelle des banques pour calculer les XVAs est d'utiliser le Least Square Monte Carlo (LSM). Cette méthode a été introduite par Longstaff et Schwartz [90] pour le calcul d'options américaines. Elle se base sur l'approximation d'espérance conditionnelle par régression. Beaucoup de travaux en finance se sont inspirés de cette approche (voir [113]). D'autres travaux se sont intéressés à la stabilité numérique de cet algorithme notamment celui de Clément, Lamberton et Protter [42].

L'adaptation du LSM aux calculs des XVAs a été introduite par Cesari et. al [23]. D'autres travaux sur le calcul des XVAs se sont inspirés de cette méthode (voir [34], [71], [70]). C'est notamment l'approche de référence pour le calcul de l'exposition sous jacente aux XVAs lorsque celle-ci ne peut être calculée par formule analytique : L'exposition, qui peut être définie comme une espérance conditionnelle, est alors approchée numériquement par régression.

1.2.1 Le LSM revu pour le calculs des XVAs

Une approche du LSM revue et adaptée aux problématiques de calcul des XVAs est introduite par Huge et Savine [76]. Cette approche est appelée *Proxies Only in*

[5] Economic capital

Indicators (POI). Elle est basée sur la série de travaux de Andreasen [16], [17], [19], [20], [18].
Pour une présentation simple de cette méthode nous nous focalisons sur le calcul de la CVA sans collatéral ni marge et recouvrement, avec un facteur d'actualisation égale à 1. Nous considérons ainsi l'équation suivante :

$$\mathrm{CVA}_0 = \mathbb{E}\left[\mathbf{1}_{0<\tau_C<T}(V_{\tau_C})^+\right], \tag{1.2.1}$$

où τ_C est le temps de défaut d'un client de la banque, et $V_t = \mathbb{E}_t\left(\int_t^T dX_s\right)$ avec X_s le processus des flux du portefeuille entre la banque et le client.
L'idée principale de cette approche est de faire non pas une LSM classique sur V, mais d'utiliser une approximation de la nonlinéarité de la CVA représentée par $\mathbf{1}_{V>0}$.
Pour cela l'équation (1.2.1) est réécrite comme suit :

$$\begin{aligned}\mathrm{CVA}_0 &= \mathbb{E}_0\left[\mathbf{1}_{0<\tau_C<T}V_{\tau_C}\mathbf{1}_{V_{\tau_C}>0}\right]\\ &= \mathbb{E}_0\left[\mathbf{1}_{0<\tau_C<T}\mathbb{E}_{\tau_C}\left(\int_{\tau_C}^T dX_s\right)\mathbf{1}_{V_{\tau_C}>0}\right]\\ &= \mathbb{E}_0\left[\int_0^T \mathbf{1}_{0<\tau_C<T}\mathbf{1}_{V_{\tau_C}>0}dX_s\right].\end{aligned}$$

Ensuite nous approximons $\mathbf{1}_{V_{\tau_C}>0}$ par $\mathbf{1}_{\widetilde{V}_{\tau_C}>0}$ où $\widetilde{V}$ est une approximation de V obtenue par régression. Nous obtenons l'approximation suivante de la CVA :

$$\mathrm{CVA}_0 \approx \mathbb{E}\left[\int_0^T \mathbf{1}_{0<\tau_C<T}\mathbf{1}_{\widetilde{V}_{\tau_C}>0}dX_s\right]. \tag{1.2.2}$$

Elle se veut plus précise que le LSM classique. En effet dans ce dernier, qui correspondrait à la formule alternative suivante

$$\mathrm{CVA}_0 \approx \mathbb{E}\left[\mathbf{1}_{0<\tau_C<T}\widetilde{V}_{\tau_C}^+\right],$$

on utilise un proxy biaisé du prix, ce qui peut créer un problème de précision, alors que dans l'approche POI on utilise les payoffs dans le calcul des XVA (avec un temps d'exercice biaisé), ce qui numériquement donne une meilleure précision : voir Figure 1.4.

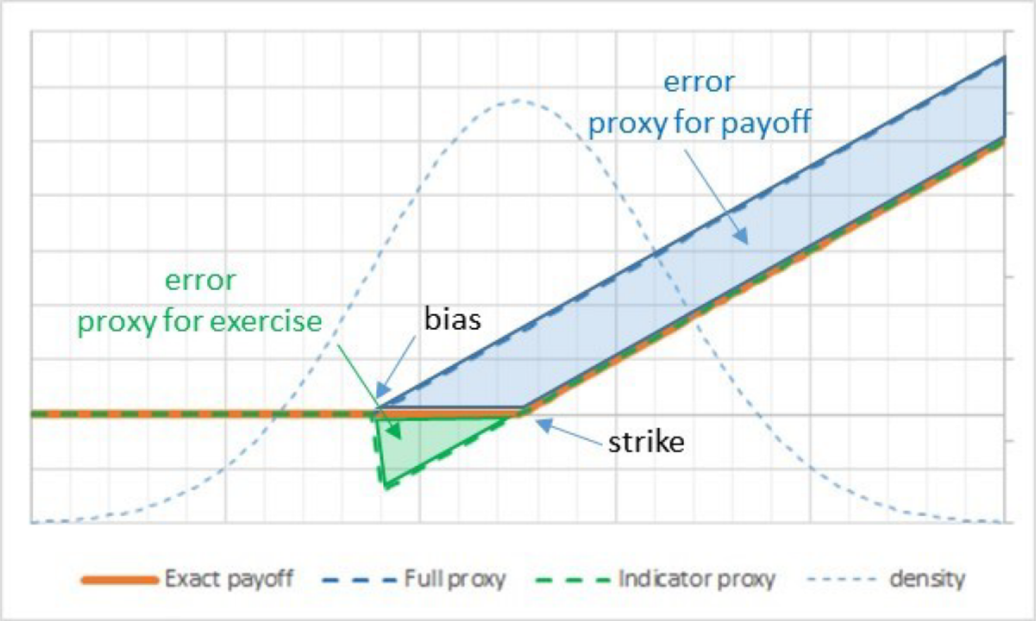


Figure 1.4 – Erreur LSM classique vs erreur POI (source Huge et Savine)

Considérons par analogie, l'exemple de calcul d'une option call sur $\widetilde{V}$ (variable aléatoire) avec un strike K. Supposons que l'approximation biaisée de V est définie par $\widetilde{V} = V + \mu$ avec μ le biais. En prenant $\widetilde{V}$ comme approximation du sous-jaçent nous utilisons un strike faux : $\max(0, \widetilde{V} - K) = \max(0, V - (K - \mu))$. Dans un cadre gaussien ceci implique une erreur de l'ordre de $\Delta\mu = \mu\Phi\left[\frac{V-K}{\sigma}\right]$ où Φ est la fonction de répartition gaussienne standard et σ est l'écart-type de V.

A contrario, avec le POI, nous utilisons le bon stike et le biais est transféré sur l'estimation de la date d'exercice. Dans ce cas on a : $\mathbf{1}_{\widetilde{V}>K}\max(0, V - K) = \max(0, V - K)\mathbf{1}_{V>K-\mu}$ et l'erreur dans un cadre gaussien est de l'ordre de
$\mu\left(\Phi\left[\frac{V-K}{\sigma}\right] - \Phi\left[\frac{V-K-\mu}{\sigma}\right]\right) \simeq \mu^2\frac{\phi(\frac{V-K}{\sigma})}{\sigma}$ où ϕ est la densité gaussienne.

1.2.2 Application de la POI au calcul de la FVA

Nous notons Φ le collatéral, modélisé comme une fonction de la valeur du portefeuille de la banque. Nous notons r^C pour le taux de rémunération du collatéral posté, r^F le taux de financement de la banque, et β_t^F, β_t^C les taux d'actualisation correspondants. Nous posons $s^F = r - r^F$ et $s^C = r - r^C$ les spreads de financement et de rémunération du collateral. Nous considérons dans la suite que les spreads s^F et s^C sont déterministes et bornés.
En partant de Elouerkhaoui [54], Proposition 1 ; nous obtenons la formulation de la FVA définie comme suit :

$$\text{FVA}_0 = \mathbb{E}\left[\int_0^T \beta_t s^C \Phi(V_t^F)dt + \int_0^T \beta_t s^F (V_t^F - \Phi(V_t^F))dt\right], \tag{1.2.3}$$

avec $V_t^F = \mathbb{E}_t\left[\int_t^T \beta_s^F \frac{dX_s}{\beta_t^F}\right]$.
Partant de cette formulation nous obtenons une représentation de la FVA comme différence d'un prix financé au taux r^F, V^F, et du prix financé au taux r, $V_t = \mathbb{E}_t\left[\int_t^T \beta_s \frac{dX_s}{\beta_t}\right]$.

Dans une première étape nous allons considérer que le colléral Φ dans l'équation (1.2.3) est nul. Dans ce cas (1.2.3) se réduit à :

$$\mathrm{FVA}_0 = \mathbb{E}\left[\int_0^T \beta_t(r_t - r_t^F)V_t^F dt\right], \tag{1.2.4}$$

Proposition 1.2.1 *Si $\Phi = 0$, alors (sous des hypothéses techniques précisées dans la preuve)*

$$FVA_0 = V_0^F - V_0. \tag{1.2.5}$$

Preuve *D'abord nous remplaçons V_t^F par sa formule dans l'equation* (1.2.4),

$$FVA_0 = \mathbb{E}\left[\int_0^T \underbrace{\beta_t s_t^F \mathbb{E}_t\left[\int_t^T \beta_s^F \frac{dX_u}{\beta_t^F}\right] dt}_{f_t}\right].$$

En supposant que $\mathbb{E}\left[\int_0^T \mid f_t \mid dt\right] < \infty$, on peut appliquer le théorème de Fubini-Tonelli pour inverser l'espérance et l'intégrale extérieures. On obtient

$$FVA_0 = \int_0^T \mathbb{E}\left[\underbrace{s_t^F \frac{\beta_t}{\beta_t^F}}_{U} \mathbb{E}_t\left[\underbrace{\int_0^T \mathbf{1}_{u>t}\beta_u^F dX_u}_{V}\right]\right] dt.$$

Comme U est G_t mesurable et bornée, $\mathbb{E}\left[U\mathbb{E}_t\left[V\right]\right] = \mathbb{E}\left[UV\right]$, ce qui donne,

$$\begin{aligned}
FVA_0 &= \mathbb{E}\left[\int_0^T s_t^F \frac{\beta_t}{\beta_t^F} \int_0^T \mathbf{1}_{u>t}\beta_u^F dX_u dt\right] \\
&= \mathbb{E}\left[\int_0^T \int_0^T \underbrace{s_t^F \frac{\beta_t}{\beta_t^F}\mathbf{1}_{u>t}\beta_u^F}_{\gamma((u,t)} dX_u dt\right]
\end{aligned}$$

Supposons que $\int_0^T \left(\int_0^T \mid \gamma(u,t) \mid^2 du\right)^{\frac{1}{2}} dt < \infty$ presque-sûrement et que X est un semimartingale, on peut appliquer le théorème de Fubini stochastique pour inverser l'intégrale de Lesbesgue et l'intégrale stochastique. On obtient

$$\begin{aligned}
FVA_0 &= \mathbb{E}\left[\int_0^T \int_0^T s_t^F \frac{\beta_t}{\beta_t^F}\mathbf{1}_{u>t} dt \beta_u^F dX_u\right] \\
&= \mathbb{E}\left[\int_0^T \int_0^u s_t^F e^{-\int_0^t s_r^F dr} dt \beta_u^F dX_u\right] \\
&= \mathbb{E}\left[\int_0^T \left(1 - e^{-\int_0^u s_r^F dr}\right) \beta_u^F dX_u\right] \\
&= \mathbb{E}\left[\int_0^T \beta_u^F dX_u\right] - \mathbb{E}\left[\int_0^T e^{-\int_0^u s_r^F dr}\beta_u^F dX_u\right] \\
&= \mathbb{E}\left[\int_0^T \beta_u^F dX_u\right] - \mathbb{E}\left[\int_0^T \beta_u dX_u\right]. \qquad \blacksquare
\end{aligned}$$

Nous allons maintenant généraliser le résultat de la Proposition 1.2.1 dans un cadre où il y a échange de collatéral[6], défini comme $\Phi(V_t^F)$. Nous réécrivons l'équation (1.2.3) comme suit :

$$\mathrm{FVA}_0 = \underbrace{\mathbb{E}\left[\int_0^T \beta_t(r_t - r_t^F)V_t^F dt\right]}_{I} - \underbrace{\mathbb{E}\left[\int_0^T \beta_t(r_t^C - r_t^F)\Phi(V_t^F)dt\right]}_{II}. \tag{1.2.6}$$

Proposition 1.2.2 *Dans le cas avec collatéral $\Phi(V_t^F)$, on a (sous des hypothèses techniques analogues à celles précisés dans la preuve de la Propostion 1.2.1)*

$$FVA_0 = \mathbb{E}\left[\int_0^T \widehat{\beta}_u dX_u\right] - V_0 \tag{1.2.7}$$

où $\widehat{\beta}_u = \beta_u^F\left(1 - \int_0^u (r_t^C - r_t^F)e^{-\int_0^t (r_w^C - r_w^F)dw}\frac{\mathbf{1}_{V_t>0}}{e^{\int_0^t s_w^C dw}}dt\right).$

Preuve *Par la Proposition 1.2.1 on a*

$$I = V_0^F - V_0.$$

De même on a

$$\begin{aligned} II &= \mathbb{E}\left[\int_0^T \beta_t(r_t^C - r_t^F)\mathbf{1}_{V_t>0}\int_0^T \mathbf{1}_{u>t}\beta_u^F\frac{dX_u}{\beta_t^F}dt\right] \\ &= \mathbb{E}\left[\int_0^T \int_0^u (r_t^C - r_t^F)e^{-\int_0^t (r_w^C - r_w^F)dw}e^{-\int_0^t s_w^C dw}\mathbf{1}_{V_t>0}dt\beta_u^F dX_u\right]. \end{aligned}$$

D'où

$$FVA_0 = I - II = \mathbb{E}\left[\int_0^T \left(1 - \int_0^u (r_t^C - r_t^F)e^{-\int_0^t (r_w^C - r_w^F)dw}e^{-\int_0^t s_w^C dw}\mathbf{1}_{V_t>0}dt\right)\beta_u^F dX_u\right] - V_0. \blacksquare$$

1.2.3 Exemples numériques

Nous présentons maintenant quelques exemples numériques. Nous considérons l'achat d'un portefeuille de call spread sur des indices FX^{ij}, où i est la monnaie domestique et j une monnaie étrangère. Nous nous intéressons à calculer la valeur de la FVA dans cette transaction.

Nous considérons un modèle HJM multifacteur pour modéliser la dynamique des taux et des changes. Pour deux économies i, j, la dynamique du taux de change FX^{ij} est donnée, pour $t > 0$, par :

$$\frac{dFX^{ji}(t)}{FX^{ji}(t)} = (r_t^i - r_t^j)dt + \langle \Gamma_{ji}^{FX}, dW_t^i\rangle, \tag{1.2.8}$$

où r_t^i, r_t^j sont les taux spot des économies i et j, Γ_{ji}^{FX} est un vecteur de volatilité déterministe et W^i est un vecteur Brownien sous la mesure $\mathbb{P}^i$ associé au numéraire $(\beta^i)^{-1}$.

Le taux spot r^i est associé au taux forward f^i par la relation $r_t^i = f^i(t,t)$, où

$$df^i(t,T) = \langle \sigma_i(t,T), \Gamma_i(t,T)\rangle dt + \langle \sigma_i(t,T), dW_t^i\rangle, \tag{1.2.9}$$

[6] Cette partie est inspirée par Moez Mrad, Head XVA desk, CACIB

avec $\sigma_i(t,T)$ et $\Gamma_i(t,T)$ des vecteurs déterministiques et W^i un vecteur Brownien sous $\mathbb{P}^i$. Pour exprimer la dynamique de r_t^j sous l'économie domestique i nous considérons un ajustement de convexité du type suivant :

$$dW_t^j = dW_t^i - \Gamma^{ji}(t)dt.$$

Nous introduisons par ailleurs les taux utilisé pour l'actualisation (ou *discount* d) : le taux risque neutre r, le taux de financement r^F et le taux de rémunération du collatéral posté r^C. Nous modélisons ces taux par des modèle de Vasicek de la forme :

$$dr_t^d = \alpha(\theta - r_t^d)dt + \langle \Gamma_t^d, dW_t^d \rangle. \tag{1.2.10}$$

Nous définissons également les spreads de financement et de rémunération du collatéral $s^F = r - r^F$ et $s^C = r - r^C$ que nous supposons constants.
La valeur du portefeuille de call spreads multidevises actualisés au taux r^d est donnée à l'instant t par :

$$V_t^d = \sum_{j=1}^{N} \left(-\mathbb{E}_t \left[e^{-\int_t^T r_u^d du} \left(FX_T^{ij} - K_1 \right)^+ \right] + \mathbb{E}_t \left[e^{-\int_t^T r_u^d du} \left(K_2 - FX_T^{ij} \right)^+ \right] \right) \tag{1.2.11}$$

avec N le nombre de monnaie, i la devise domestique, K_1 et K_2 les strikes de l'option, T la maturité.

Comme tous les processus sont modélisés dans un cadre gaussien nous pouvons calculer V_t^d par formule analytique. Nous rappelons la formulation (1.2.4)-(1.2.5) de la FVA d'un call spread de change FX^{ij} dans un cadre sans collatéral :

$$\mathrm{FVA}_0 = \mathbb{E} \left[\int_0^T \beta_t (r_t - r_t^F) V_t^F dt \right] = V_0^F - V_0,$$

où V correspond à V^d pour $r^d = r$ dans (1.2.11).

La figure 1.5 montre l'évolution de l'erreur relative de la FVA (sans collatéral) calculer par (1.2.4) et (1.2.5) sur différentes maturités du portefeuille. Cette erreur est donnée en fonction du nombre de trajectoires de MC. Nous constatons que la maturité du portefeuille joue un rôle important sur la convergence de la méthode MC. Pour une maturité courte ($T = 0.2$) on a une erreur relative de l'ordre de 0.01 en générant 10^3 scénarios de MC alors que pour une maturité plus longue ($T = 1$) nous obtenons une erreur relative de l'ordre de 0.2 pour le même nombre de scénarios.

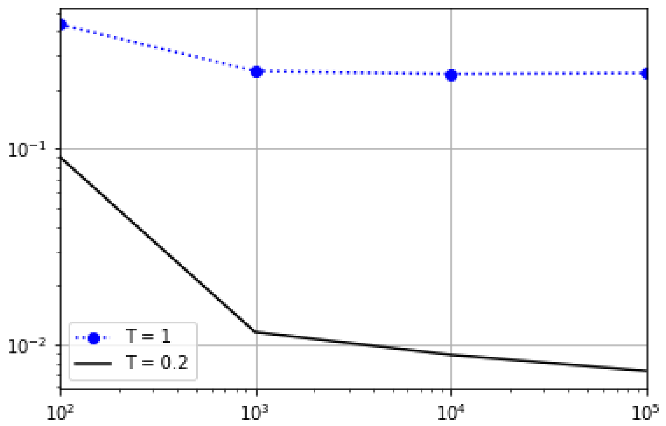


FIGURE 1.5 – Erreur relative de la FVA (sans collateral) en fonction du nombre de trajectoire de MC : $N = 2$, $FX_0 = 1$, $r_0 = 0.1$, $s^F = 0.001$, $K_1 = 0.95$, $K_2 = 1.2$.

La figure 1.6 montre l'évolution de l'erreur relative de la FVA (sans collatéral) calculer par (1.2.4) et (1.2.5) sur des portefeuilles avec différents nombres de devises. Cette erreur est donnée en fonction du nombre de trajectoires de MC. On observe que le nombre de devises affecte la convergence de l'algorithme.

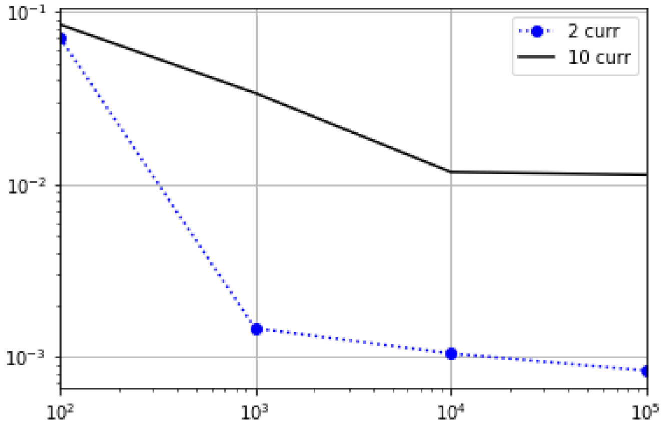


FIGURE 1.6 – Erreur relative de la FVA (sans collateral) en fonction du nombre de trajectoire MC. $FX_0 = 1, r_0 = 0.1, s^F = 0.001, K_1 = 0.95, K_2 = 1.2$, T = 0.2.

Nous considérons dans la suite un portefeuille call spreads de deux devises. Dans un cadre avec collateral, la FVA s'écrit alors comme suit :

$$\begin{aligned}\mathrm{FVA}_0 &= \mathbb{E}\left[\int_0^T \beta_t(r_t - r_t^F)V_t^F - \beta_t(r_t^C - r_t^F)\Phi(V_t^F)dt\right],\\ &= \mathbb{E}\left[\widehat{\beta}\sum_{j=1}^{N}\Big(-\big(FX_T^{ij} - K_1\big)^+ + \big(K_2 - FX_T^{ij}\big)^+\Big)\right] - V_0,\end{aligned}$$

où

$$\widehat{\beta} = \beta_T^F\left(1 - \int_0^T (r_t^C - r_t^F)e^{-\int_0^t (r_w^C - r_w^F)dw}\frac{\mathbf{1}_{V_t>0}}{e^{\int_0^t s_w^C dw}}dt\right).$$

Nous utilisons la méthode POI, approximant l'indicatrice $\mathbf{1}_{V_t>0}$ qui apparait dans $\widehat{\beta}$ par $\mathbf{1}_{\widetilde{V}_t>0}$ où $\widetilde{V}$ est une approximation de V par régression. Ainsi la FVA est approximée par :

$$\mathrm{FVA}_0 \approx \mathbb{E}\left[\widehat{\beta}\sum_{j=1}^{N}\left(-\left(FX_T^{ij}-K_1\right)^+ + \left(K_2 - FX_T^{ij}\right)^+\right)\right] - V_0, \qquad (1.2.12)$$

où $\widehat{\beta} \approx \beta_T^F \left(1 - \int_0^T (r_t^C - r_t^F) e^{-\int_0^t (r_w^C - r_w^F)dw} \frac{\mathbf{1}_{\widetilde{V}_t>0}}{e^{\int_0^t s_w^C dw}} dt\right)$.

La Figure 1.7 montre l'erreur relative de la FVA avec collateral calculer en utilisant la méthode POI (Figure de Gauche) et en utilisant la méthode LSM (Figure de Droite). Cette erreur est donnée en fonction du nombre de trajectoires de MC. Nous utlisons l'ensemble des facteurs de risque et leur carré comme base de régression. On observe que la méthode POI a une erreur relative plus petite que celle par LSM. Cependant cette méthode souffre du problème de grande dimension et de longues maturités.

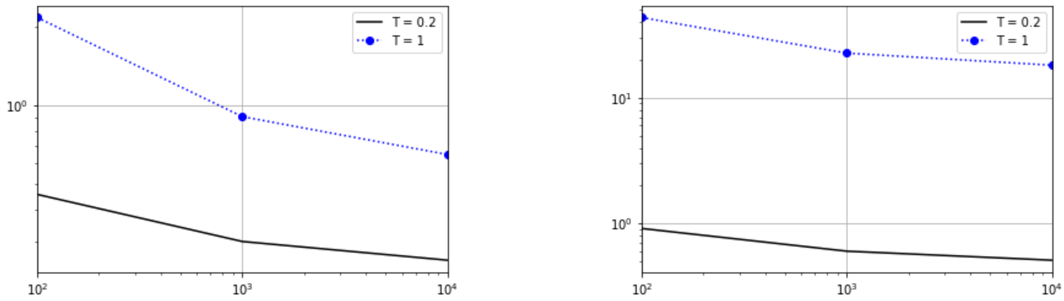


FIGURE 1.7 – Erreur relative de la FVA (avec collateral) en fonction du nombre de trajectoire MC : $FX_0 = 1, r_0 = 0.1, K_1 = 0.95, K_2 = 1.2, s_F = 10bps, s_C = 5bps$. Gauche : Régression par POI ; Droite : Régression classique.

Les méthodes numériques basées sur la régression (LSM, POI) sont bien adaptées pour le calcul des XVAs de première génération (CVA, FVA). Cependant, les XVAs de seconde génération (MVA, KVA) impliquant le calcul de mesures de risque conditionnelles dans le futur nécessitent d'explorer d'autres méthodes. Cette thèse traite de méthodes numériques basées sur le Nested Monte Carlo (NMC) pour le calcul des XVAs.

1.3 Contribution de cette thèse

Dans un premier travail nous considérons le problème pour une banque (ou une assurance) du calcul numérique de son capital économique sous forme d'une $value-at-risk$, ou d'une *expected shortfall*, de sa perte sur un horizon de temps donné. Cette perte inclut l'appréciation du modèle d'évaluation du passif de l'entité, que nous estimons par un Monte Carlo imbriquée à la Gordy et Juneja [68] ou par une régression à la Broadie, Du, et Moallemi [37]. En utilisant une approche d'approximation stochastique sur la value-at-risk ou l'expected shortfall, nous établissons la convergence des schémas résultant de la simulation du capital économique, sous des hypothèses modérées ne portant que sur le problème de limite théorique, par opposition aux hypothèses sur les problèmes d'approximation faites dans [68] et [37]. Nos estimations de capital économique peuvent ensuite être considérées conditionnellement dans un cadre Markovien et intégrées dans une simulation de Monte Carlo externe, afin de générer une *market value margin* (MVM) en assurance ou un *capital valuation adjustment* (KVA) dans le langage bancaire. Ceci est illustré numériquement par une étude de cas de calcul de KVA mis en oeuvre sur des GPU [7].

Dans un second travail, nous présentons une application du NMC au calcul des XVAs par un algorithme parallèle adapté au GPU. Le calcul global des XVA peut impliquer jusqu'à cinq niveaux de couche de Monte Carlo. Les couches les plus hautes sont d'abord lancées et déclenchent si nécessaire des simulations imbriquées pour calculer un élément à une couche inférieure. Si l'utilisateur n'est intéressé que par certaines composantes de XVAs, seul le sous-arbre correspondant au XVA le plus externe doit être calculé numériquement. De plus les couches internes n'ont besoin que d'un nombre racine carrée de simulations par rapport à la couche la plus externe. Enfin, certaines des couches présentent des constantes de variance plus faible. En conséquence, avec les GPU, de tels calculs NMC XVA sont réalisables et garantissent un contrôle de l'erreur commise. Cependant, bien que le NMC soit en principe adapté à la parallélisation, une implémentation GPU efficace des calculs NMC XVA nécessite diverses optimisations. Ceci est illustré dans des exemples de calculs de couches de XVAs impliquant des dérivés actions, taux d'intérêt et crédit.

Dans un troixième travail, nous présentons un nouvel algorithme basé sur un Monte Carlo imbriqué à une couche (1NMC) pour simuler les fonctions U d'un processus de Markov X. La principale originalité de la méthode proposée provient du fait qu'elle fournit une recette pour simuler $U_{t\geq s}$ conditionnellement à X_s. Cette méthode peut être utilisée pour un grand nombre de problèmes, notamment : les équations différentielles stochastiques inversées (BSDE), les réfléchies BSDE (RBSDE), les mesures de risque et au-delà. La généralité, la stabilité et le caractère itératif de cet algorithme, même en haute dimension, en font sa force. Il est bien sûr plus lourd qu'un Monte Carlo (MC), mais il est beaucoup plus précis de simuler des quantités presque impossibles à simuler avec MC. De plus, l'adéquation parallèle de 1NMC le rend possible dans un temps de calcul raisonnable. Nous expliquons cet algorithme, étudions sa convergence ainsi que sa stabilité et sa complexité. Nous fournissons également divers exemples numériques de grandes dimensions (100) exécutés en quelques minutes sur un processeur graphique (GPU).

Nous allons dans la suite de l'introduction présenter les principales contributions de ce travail.

1.3.1 Algorithme Stochastique pour le calcul du capital économique

Le calcul de la KVA pour une banque (où de la MVM pour une assurance) nécessite l'évaluation de son capital économique future. Nous modélisons le capital

[7] Graphics Process Unit

économique future comme une expected shortfall ES_t conditionnelle à un instant t de la perte de la banque L sur un horizon de risque futur.

Etant donnée l'évolution d'un facteur de risque (Z_t), nous définissons la perte comme une variable aléatoire de la forme

$$L = \phi + \beta\, \mathbb{E}_0 \left[\psi | Z_1\right] - \mathbb{E}_0 \left[\psi'\right], \tag{1.3.1}$$

avec $(\beta, \phi, \psi, \psi')$ des variables aléatoires. Nous notons par $\mathrm{P}(z, \cdot)$ et $\mathrm{Q}(z, \cdot)$ la distribution de Z_1 et L conditionnellement à $Z_0 = z$. Les deux derniers termes de (1.3.1) modélisent les passifs futurs et présents, alors que le premier terme correspond à la perte de trading réalisée par la banque sur l'intervalle [0,1].

Comme établi par Rockafellar et Uryasev [109], [110], la value-at-risk $\xi_\star$ au seuil α de L est caractérisée par l'équation :

$$1 - \frac{1}{1-\alpha}\mathbb{P}_0 \left(L > \xi_\star\right) = \mathbb{E}_0 \left[H_1(\xi_\star, L)\right] = 0, \tag{1.3.2}$$

où

$$H_1(\xi, x) = 1 - \frac{1}{1-\alpha}\mathbf{1}_{x>\xi}.$$

Etant donnée une solution $\xi_\star$ de (1.3.2), l'expected shortfall $\chi_\star$ correspondant est la solution de l'équation

$$\chi_\star - \xi_\star - \frac{1}{1-\alpha}\mathbb{E}_0 \left[(L - \xi_\star)^+\right] = \mathbb{E}_0 \left[H_2(\xi_\star, \chi_\star, L)\right] = 0 \tag{1.3.3}$$

où

$$H_2(\xi, \chi, x) = \chi - \xi - \frac{1}{1-\alpha}\left(x - \xi\right)^+.$$

Notre capital économique est défini comme $\chi_\star$.

L'algorithme stochastique est une procédure itérative introduite par Robbins et Monro [107] et Kiefer et Wolfowitz [82] permettant de trouver la racine d'équations telles que (1.3.2)-(1.3.3) fonction non connue H.

Soit une suite de vecteurs aléatoires $(\theta_n)_{n\geqslant 1}$ à valeurs dans un espace euclidien de dimension finie, l'agorithme stochastique pour résoudre $\mathbb{E}_0(H(\theta, \epsilon)) = 0$ est la procédure définie récursivement par,

$$\theta_{k+1} = \theta_k + \gamma_{k+1} H(\theta_k, \epsilon_k), \tag{1.3.4}$$

où $(\gamma_k)_{k\geqslant 1}$ est le pas de l'algorithme et ϵ_k une variable aléatoire indépendant de même loi que ϵ. Les résultats classiques classique de l'algorithme stochastique (voir [51], [83], [32]) permettent d'établir la convergence et la vitesse de convergence de l'algorithme.

Pour la résolution numérique des équations (1.3.2), (1.3.3), Bardou, Pagès et Frikha [28] proposent une procédure basée sur une combinaison de deux algorithmes stochastiques pour le calcul du couple $(\mathrm{VaR}, \mathrm{ES})$ dans un cadre où L est i.i.d. Cette procédure est définie par :

$$\begin{cases} \xi_{k+1} &= \xi_k - \gamma_{k+1} H_1(\xi_k, L_{k+1}), \\ \chi_{k+1} &= \chi_k - \gamma_{k+1} H_2(\xi_k, \chi_k, L_{k+1}). \end{cases} \tag{1.3.5}$$

avec une séquence déterministique de pas de l'algorithme $\{\gamma_k, k \geq 1\}$ et une séquence de variables aléatoires i.i.d $\{L_k, k \geq 1\}$ de loi $\mathrm{Q}(z, \cdot)$. L'étude de l'algorithme permet d'établir une convergence presque sûre et une convergence en loi sous des hypothèses sur les séquences de pas de l'algorithme $(\gamma_k)_{k\geqslant 1}$ et sur les fonctions H_1 et H_2. Plus précisément, ces convergences sont établies sous les hypothéses suivantes :

H1 $\{\gamma_k, k \geq 1\}$ *est un séquence deterministe de* $(0,1)$*-valeur, tels que sur* $\kappa \in (0,1]$,

$$\sum_k \gamma_k = +\infty, \qquad \sum_k \gamma_k^{1+\kappa} < +\infty.$$

H2 *a) Sous* $\mathbb{P}_0$*,* $L = \phi \sim \mathrm{Q}(Z_0 = z, \cdot)$ *a une distribution continue.*
b) $\mathbb{E}_0\left[L^2\right] = \int x^2 \mathrm{Q}(z, \mathrm{d}x) < +\infty$.

Theorem 1.3.1 *Supposons H1 and H2. Alors il existe une variable aléatoire bornée* $\xi_\star$ *et un nombre réel* $\chi_\star$ *solutions de* (1.3.2)-(1.3.3) *tel que, pour tout* $p \in (0,2)$*, presque sûrement,*

$$\mathbb{P}_0\left(\lim_{k\to\infty}(\xi_k, \chi_k) = (\xi_\infty, \chi_\infty)\right) = 1, \qquad \lim_{k\to\infty} \mathbb{E}_0\left[|\xi_k - \xi_\infty|^p\right] = 0.$$

Une des contributions de cette partie est d'étendre l'algorithme 1.3.5 dans un cadre où L est définie par l'équation 1.3.1. En effet pour le calcul du deuxième terme de (1.3.1) nous appliquons deux approches :

M1 - Calculer $\mathbb{E}_0\left[\psi | Z_1\right]$ en regénérant des trajectoires à partir du temps 1 (NMC). Cette méthode est l'approche de Gordy et Juneja [68].

M2 - Calculer $\mathbb{E}_0\left[\psi | Z_1\right]$ par une régression empirique à l'instant 1. Cette méthode est l'approche de Broadie, Du et Moallemi [37].

Le calcul de $\mathbb{E}_0\left[\psi'\right]$ peut se faire par un Monte Carlo standard.
Nous établissons ainsi les hypothèses nécessaires pour étendre les résultats du théorème 1.3.1 aux cas **M1** et **M2**. Au final, cette procédure sera conditionnée à des simulations extérieures de MC afin de calculer la KVA.

1.3.2 XVA, NMC et GPU optimisation

Dans cette partie nous proposons une méthode numérique basée sur le NMC pour le calcul des XVAs.
La simulation imbriquée se pésente sur des problèmes impliquant l'approximation d'une fonction conditionnelle tel que la valorisation d'un portefeuile à une date future (Gordy et Juneja [68]).

Nous voulons calculer numériquement I donné par :

$$I = \mathbb{E}\left[f(y, g(y))\right], \tag{1.3.6}$$

où f est une fonction quelconque, y une variable aléatoire suivant la loi $\mu(y)$ et

$$g(y) = \mathbb{E}\left[\Phi(y, Z)\right],$$

dans lequel Φ une fonction quelconque et Z suit la loi $\mu(z|y)$.

Une méthode de MC classique ne permet pas d'avoir une approximation de (1.3.6) à cause de l'espérance conditionnelle. Le NMC résoud cela en estimant l'espérance conditionnelle interne par une moyenne arithmétique : Une première couche de trajectoires de MC est utilisée pour estimer l'espérance de (1.3.6) et une deuxième couche imbriquée (conditionnelle à la première) est générée pour estimer l'espérence conditionnelle. Ainsi nous estimons (1.3.6) par :

$$I \approx \frac{1}{M} \sum_{i=1}^{M} \left[f(y_i, \frac{1}{N} \sum_{n=1}^{N} \Phi(y_i, z_i^n)) \right], \tag{1.3.7}$$

où M est le nombre de trajectoires extérieures et N le nombre de trajectoires intérieures avec y_i et z_i^n des variables aléatoires suivant respectivement les lois $\mu(y)$ et $\mu(z|y)$. Dans Gordy et Juneja [68] une relation est donnée entre le nombre de trajectoire extérieur et intérieur afin de minimiser le biais et la variance de cet estimateur.

Dans ce travail nous proposons un arbre de NMC pour le calcul des XVAs (voir Figure 3.1 (en Sect. 3.3.2)). L'ensemble des XVAs pourrait ainsi être calculé par NMC avec un nombre de couches dépendant du niveau d'imbrication de la XVA correspondant.

Nous proposons ainsi l'algorithme XVA NMC suivant :

```
Input : Le portefeuille de la banque, les courbes de crédit de la
  banque et des clients, et, possiblement, une nouvelle transaction ;
Choisir les couches dans le sous-arbre de la Figure 3.1 (de
  Sect. 3.3.2), avec les nombres de simulation correspondant, notés
  $M_{(0)}, \dots, M_{(i)}$, pour $1 \leq i \leq 5$ (nous supposons au moins un niveau
  de simulation imbriqué) ;
Par dichotomie sur $M_{(0)}$, atteindre une erreur acceptable (dans le
  sens de l'intervalle de confiance extérieur) pour
  $M_{(0)} \otimes M_{(1)} \dots \otimes M_{(i)}$ NMC avec $M_{(1)} = \dots = M_{(i)} = \sqrt{M_{(0)}}$ ;
Pour chaque $j$ décroissant de $i$ à 1, atteindre par dichotomie $M_{(j)}$
  un biais ciblé (dans le sens de l'intervalle de confiance extérieur)
  pour $M_{(0)} \otimes M_{(1)} \otimes \dots \otimes M_{(j)} \otimes \dots \otimes M_{(i)}$ NMCs ;
Return(Les métriques au temps 0 relatives au portefeuille de la banque, et
  le cas échéant, les métriques incrémentales liés à une transaction
  provisoire.)
```

Algorithm 0: XVA NMC algorithm.

Une partie importante de ce travail est de proposer une implémentation parallèle optimale de l'algorithme 0. Pour cela nous proposons les différentes optimisations parallèles nécessaire dans le cadre des XVAs et nous testons ces algorithmes sur GPU grâce à la programmation CUDA.

1.3.3 Apprentissage de Monte Carlo conditionel pour les diffusions

La simulation Monte Carlo est largement considérée comme la méthode de référence pour résoudre les problèmes paraboliques linéaires à haute dimension. Dans le cas non-linéaire, le Monte Carlo (à une seule couche de trajectoires tout au moins) souffre de la malédiction de la dimension. Tout ou presque tout a été dit sur les astuces numériques à une seule couche nécessaires pour forcer la convergence de Monte Carlo dans les cas non-linéaires.

Soient $(X_t)_{t\in[0,T]}$ un processus G$_t$-Markov à valeurs dans $\mathbb{R}^{d_1}$ et la discrétisation temporelle $\{t_0, ..., t_{2^L}\} = \{0, T/2^L, ..., T\}$. Nous présentons un nouvel algorithme basé sur une simulation à une imbrication (One-layered Nested) de MC à deux couches pour approximer la fonctionnelle U de X définie, pour tout $s \in \{t_0, ..., t_{2^L}\}$, par

$$(f) \qquad U_s = \mathbb{E}_s\left(\sum_{t_k \geq s}^{2^L} f(t_k, X_{t_k}, X_{t_{k+1}})\right) = \mathbb{E}\left(\sum_{t_k \geq s}^{2^L} f(t_k, X_{t_k}, X_{t_{k+1}})\middle| \mathrm{G}_s\right),$$

où $\mathbb{E}_s(\cdot) = \mathbb{E}\left(\cdot \middle| \mathrm{G}_s\right)$ est l'espérance sous la mesure P, chaque fonction déterministe $f(t_k, \cdot, \cdot)$ est $\mathcal{B}(\mathbb{R}^{d_1}) \otimes \mathcal{B}(\mathbb{R}^{d_1})$-mesurable et satisfait la condition de carré intégrabilité $\mathbb{E}(f^2(t_k, X_{t_k}, X_{t_{k+1}})) < \infty$ avec la convention $f(t_{2^L}, X_{t_{2^L}}, X_{t_{2^L+1}}) = f(t_{2^L}, X_{t_{2^L}})$. Plusieurs problèmes numériques fondés sur la résolution d'équation différentielle stochastique rétrograde (EDSR) et exercice optimal (EO) peuvent s'écrire sous la forme donnée dans (f). De ce fait, nous allons expliquer l'ensemble de la méthode pour la simulation de U puis expliquer l'étendue des applications pour EDSR, EO et mesures de risques.

Lorsque la majorité des contributions actuelles ciblent l'estimation de U_{t_k} pour $k = 0, ..., 2^L$ connaissant la réalisation $\{X_{t_j}\}_{0 \leq j \leq k}$, notre objectif est de simuler des approximations $\{U_s^{m_0,m_1}\}_{s \geq t_{k+1}}$, avec $(m_0 = 1, ..., M_0)$ et $(m_1 = 1, ..., M_1)$, de $\{U_s\}_{s \geq t_{k+1}}$ conditionnellement à la réalisation $\{X_{t_j}^{m_0}\}_{0 \leq j \leq k}$. Cette tâche requiert la simulation et la sauvegarde en mémoire d'une première couche $(X^{m_0})_{m_0=1,\ldots,M_0}$ de trajectoires, puis une seconde $(X^{m_0,m_1})_{m_1=1,\ldots,M_1}$ couche de trajectoires non sauvegardées utilisées à l'apprentissage des approximations $\{U_s^{m_0,m_1}\}_{s \geq t_{k+1}}$. Malgré qu'elle soit plus complexe, cette procédure procure beaucoup plus de possibilités. En particulier, elle permet la simulation de U_{t_k} en calculant la moyenne

$$\frac{1}{M_1} \sum_{m_1=1}^{M_1} \left(f(t_k, X_{t_k}^{m_0}, X_{t_{k+1}}^{m_0,m_1}) + U_{t_{k+1}}^{m_0,m_1}\right).$$

Connaissant aussi des réalisations $U_s^{m_0,m_1}$, il est possible de calculer des quantiles ou, plus remarquable encore, de simuler un processus $\widetilde{U}$ défini par $(\widetilde{f})$ ($\widetilde{f}$ remplace f dans l'équation (f)) avec une fonction $\widetilde{f}$ dépendante de U comme par exemple $\widetilde{f}(t_k, x, y) = f(t_k, U_{t_k}(x), U_{t_{k+1}}(y))$. Ce dernier cas de figure est très souvent rencontré dans la simulation de la X-Valuation Adjustment.

Bien que nous ne soyons pas les premiers à proposer l'apprentissage pour les EDSR et EO, nous sommes les premiers à le faire via un Monte Carlo imbriqué. Sur cette base de Monte Carlo imbriqué, nous proposons aussi une méthode itérative permettant d'augmenter la précision en ajoutant plus d'étapes et donc en augmentant la profondeur de l'apprentissage. Utilisant l'égalité

$$\mathbb{E}(U_s) = \mathbb{E}\left(U_{s'} + \sum_{t_{l+1} > s}^{s'} f(t_l, X_{t_l}, X_{t_{l+1}})\right)$$

vraie pour $s' > s$ et sa forme localisée pour chaque intervalle $[a, b]$

$$\mathbb{E}\big(U_s 1_{\{U_s \in [a,b]\}}\big) = \mathbb{E}\left(1_{\{U_s \in [a,b]\}} \left[U_{s'} + \sum_{t_{l+1} > s}^{s'} f(t_l, X_{t_l}, X_{t_{l+1}})\right]\right),$$

nous présentons aussi une méthode non-paramétrique d'estimation et de contrôle efficace du biais. De la même manière, nous détaillons une méthode d'ajustement de variance fondée sur l'égalité

$$\mathbb{E}\left(\mathrm{Var}_s(U_{s'})\right) = \mathbb{E}\left(\mathbb{E}_s\left([U_{s'} - \mathbb{E}_s(U_{s'})]^2\right)\right) = \mathbb{E}\left([U_{s'} - \mathbb{E}_s(U_{s'})]^2\right).$$

vraie pour $s' > s$ et sa forme localisée pour chaque intervalle $[a, b]$

$$\mathbb{E}\big(\mathrm{Var}_s(U_{s'})1_{\{\mathrm{Var}_s(U_{s''})\in[a,b]\}}\big) = \mathbb{E}\big(1_{\{\mathrm{Var}_s(U_{s''})\in[a,b]\}}[U_{s'} - \mathbb{E}_s(U_{s'})]^2\big)$$

vraie pour $s' > s$ et $s'' > s$. L'ajustement de variance proposé permet une simulation imbriquée des queues de distribution sans faire appel à des méthodes d'échantillonnage préférentiel. La bonne représentation des queues, via cet ajustement, devient incontournable pour certains problèmes non-linéaires comme par exemple l'EO.

Chapitre 2

Stochastic approximation schemes for economic capital and risk margin computations

This chapter is based on Barrera, et al., [30].

2.1 Introduction

The current financial and insurance regulatory trends incentivize investment banks and insurance companies to charge to their clients, on top of a risk-neutral expectation of contractual cash flows, a suitable risk margin (see [96], [97]), meant to be gradually released to shareholders as return for their capital at risk in the future. This risk margin, sometimes called market value margin (MVM) in insurance and corresponding in banking to a capital valuation adjustment (KVA, see [12]), can be modeled as an expectation of the future economic capital of the firm. Future economic capital is modeled in our paper as the conditional expected shortfall (ES)[1] of the losses of the firm over a one-year horizon. These losses are assessed on a mark-to-model basis, which includes, at any future time point where the conditional expected shortfall is computed, the valuation one year later of the liabilities of the firm, such as variable annuities (VA) in the insurance case or a credit valuation adjustment (CVA) in the banking case, i.e. an expectation, conditionally on the information available one year later, of the future cash flows that the liability is pricing.

As such complex liabilities are typically intractable analytically and because the losses of the firm are specified through dynamic models of the underlying risk factors, in principle, the computation of the risk margin involves a nested and, in fact, a doubly nested simulation, whereby an outer Monte Carlo simulation gathers inner estimates of conditional expected shortfalls at future time points, themselves calling for recursive valuation one year later of the embedded liability. This makes it a challenging problem, both from a practical and from a convergence analysis point of view. In particular, on realistically heavy applications, such computations can only be implemented in parallel, with GPUs as a current hardware paradigm, which poses nontrivial programming optimization issues.

The assumptions made in Gordy and Juneja [68] for establishing the convergence of the simulation-and-sort value-at-risk and expected shortfall nested Monte Carlo estimates are hard to check (and might actually be violated) in practice, especially when considered dynamically in the context of risk margin computations. As the value-at-risk and expected shortfall of a given loss random variable can jointly be represented as zeros of suitable functions that can be written as expectations, an alternative is stochastic approximation (SA). In the base case without embedded

[1] In the context of this paper where we are considering conditional ES, we avoid the alternative terminology of conditional value-at-risk for (unconditional) ES.

liability of the firm, the convergence of the value-at-risk and expected shortfall SA estimates is established in Bardou, Frikha, and Pagès [28], [29]. In the present paper this convergence is extended to the case of dependent noise, corresponding to the presence of the nested future liability of the firm in our loss variable. This is then applied to risk margin computations by embedding the resulting inner conditional ES estimates into an outer sample mean.

Moreover we analyze a variant of this approach where the future liabilities are regressed as in Broadie, Du, and Moallemi [37], rather than re-simulated in a nested fashion, resulting in a simply nested procedure for the overall risk margin computation.

The different variants of the method are tested numerically, using GPU programming so that the inner conditional risk measures can be computed in parallel and then averaged out for yielding the outer risk margin estimate.

Beyond the extension of the base result of [28, 29] to dependent noise and its economical capital and risk margin application, we refer the reader to the concluding section of the paper regarding the technical contributions of our approach with respect to [68] and [37].

The paper is organized as follows. Section 2.2 presents our stochastic approximation value-at-risk and expected shortfall algorithms in the presence of dependent noise, with nested Monte Carlo versus regression estimates of the latter in the respective cases of Algorithms 2 and 3 (whereas Algorithm 1 corresponds to the base case without dependent noise). Sections 2.3 and 2.4 deal with the convergence analyses of the respective Algorithms 2 and 3. Section 2.5 casts such estimates in a dynamic setup, integrating out the estimated conditional economic capital in the context of an outer simulation for the corresponding risk margin ; this is then illustrated numerically in the context of a KVA case study. Section 2.6 concludes.

Remark 2.1.1 *In the motivating discussion above and in our application Section 2.5, for concreteness, we focus on economic capital, modeled as expected shorfall, and on the ensuing risk margin. However, the results of Sections 2.3 and 2.4 cover both expected shorfall and value-at-risk (establishing convergence for the latter is in fact a prerequisite for the former). Hence, our results also cover the cases of value-at-risk, conditional value-at-risks, and integration of the latter in the context of an outer expectation. Again this can be relevant together for bank and for insurance, noting that :*

- *In the insurance case, Solvency capital is determined as the 99.5%-value-at-risk of the one year loss of the firm for Solvency II (see [96]), and as the 99%-expected shortfall for the Swiss Solvency Test (see [97]) ;*
- *In the banking case, Basel II Pillar II defines economic capital as the 99% value-at-risk of the depletion over a one-year period of core equity tier I capital (CET1) (where the latter corresponds the one year trading loss of the bank as detailed in [12, Section A.2]) ; But the FRTB required a shift from 99% value-at-risk to 97.5% expected shortfall as the reference risk measure in capital calculations. Moreover, value-at-risk is relevant to banks for the computation of their initial margin (with a time horizon of one or two weeks, as opposed to one year conventionally in the paper) and, in turn, of their dynamic (conditional) initial margin (see [21]) in the context of the computation of their margin valuation adjustment (MVA).*

2.2 Stochastic Algorithms for Economic Capital Calculations

On some probability space $(\Omega, \mathcal{A}, \mathbb{P})$, our financial loss L is defined as a real valued random variable of the form

$$L = \phi + \beta \, \mathbb{E}_0 \left[\psi | Z_1 \right] - \mathbb{E}_0 \left[\psi' \right], \tag{2.2.1}$$

where $(\beta, \phi, \psi, \psi')$ are four real valued random variables and (Z_0, Z_1) are two $\mathbb{R}^q$ valued random variables such that under $\mathbb{P}_0$, the conditional probability measure $\widehat{\mathbb{P}}$ given Z_0 (with related expectation and variance denoted by $\mathbb{E}_0$ and $\mathbb{V}\mathrm{ar}_0$) :

— i.i.d. samples from (ϕ, β, Z_1) given Z_0 are available ;

— i.i.d. samples from the conditional distribution of ψ given Z_1, denoted by $\Pi(Z_1, \cdot)$, are available ;

— i.i.d. samples from the conditional distribution of ψ' given $\{Z_0 = z\}$, denoted by $\Pi'(z, \cdot)$, are available ;

— the discount factor β is bounded : there exists a positive constant c_β such that $|\beta| \leq c_\beta$.

We denote by $\mathrm{P}(z, \cdot)$ and $\mathrm{Q}(z, \cdot)$ the distributions of Z_1 and L conditionally on $Z_0 = z$. We also write

$$\Psi(Z_1) := \mathbb{E}_0[\psi | Z_1] \text{ and } \Psi'(z) := \mathbb{E}_0[\psi'], \text{ so that } L = \phi + \beta \Psi(Z_1) - \Psi'(z). \tag{2.2.2}$$

In the financial application the second and third terms in (2.2.1) will be used for modeling the future (conventionally taken as 1, i.e. one year) and present (time 0) liability valuations, whereas the first term corresponds to the realized loss of the firm on the time interval $[0, 1]$. The above-listed assumptions allow recovering $\mathbb{E}_0 \left[\psi | Z_1 \right]$ by nested Monte Carlo simulation restarting from time 1 (which is the approach in [68]) or by empirical regression at time 1 (which is the approach in [37]), whereas $\mathbb{E}_0 \left[\psi' \right]$ can be obtained by a standard Monte Carlo simulation rooted at $(0, z)$.

Let

$$H_1(\xi, x) := 1 - \frac{1}{1-\alpha} \mathbf{1}_{x > \xi}, \; H_2(\xi, \chi, x) := \chi - \xi - \frac{1}{1-\alpha} \left(x - \xi \right)^+ . \tag{2.2.3}$$

A value-at-risk $\xi_\star$ at level α of the random variable (loss) L solves the equation

$$1 - \frac{1}{1-\alpha} \mathbb{P}_0 \left(L > \xi_\star \right) = \mathbb{E}_0 \left[H_1(\xi_\star, L) \right] = 0; \tag{2.2.4}$$

it is uniquely defined if L has an increasing $\mathbb{P}_0$ c.d.f F e.g. if it has a nonvanishing $\mathbb{P}_0$ density f. Given a solution $\xi_\star$ to (2.2.4), the expected shortfall $\chi_\star$ at level α solves the equation

$$\chi_\star - \xi_\star - \frac{1}{1-\alpha} \mathbb{E}_0 \left[(L - \xi_\star)^+ \right] = \mathbb{E}_0 \left[H_2(\xi_\star, \chi_\star, L) \right] = 0 \tag{2.2.5}$$

(noting that any solution $\xi_\star$ to (2.2.4) yields the same equation (2.2.5) for $\chi_\star$; see e.g. Lemma 2.7.1 in Appendix 2.7.3). Equivalently, $\xi_\star$ and $\chi_\star$ satisfy

$$\int_{\xi_\star}^{+\infty} \mathrm{Q}(z, dx) = 1 - \alpha, \qquad \chi_\star = \xi_\star + \frac{1}{1-\alpha} \int (x - \xi_\star)^+ \mathrm{Q}(z, \mathrm{d}x). \tag{2.2.6}$$

We model economic capital (EC) at time 0 (known in the insurance regulation as the Solvency capital requirement, SCR) as the expected shortfall of level $\alpha \in \left(\frac{1}{2}, 1\right)$ of the distribution of L given $Z_0 = z$, i.e.

$$\mathrm{ES}(z) := (1-\alpha)^{-1} \int_\alpha^1 \mathbb{V}\mathrm{a}\mathbb{R}_0^a[L]\, \mathrm{d}a. \tag{2.2.7}$$

In (2.2.7), $\mathbb{V}\mathrm{a}\mathbb{R}_0^a[L]$ is a corresponding value-at-risk at level a. Throughout the paper, α is fixed, so the dependence of $\mathrm{ES}(z)$ upon α is omitted. Likewise we introduce the notation $\mathrm{VaR}(z)$ for the value-at-risk at the (fixed) level α of l.

2.2.1 Stochastic Approximation (SA) With Dependent Noise

We propose two approaches for computing $\mathrm{ES}(z)$. Both estimates $\widehat{\mathrm{ES}(z)}$ are defined as the output of a stochastic approximation (SA) algorithm with K iterations. However, in the applications targeted in this paper, the expectations in (2.2.4) and (2.2.5) are not known analytically, so that the quantities $(\xi_\star, \chi_\star)$ are roots of intractable functions. SA algorithms provide a numerical solution to (2.2.4)-(2.2.5) (see e.g. [32], [83]) : given a deterministic stepsize sequence $\{\gamma_k, k \geq 1\}$ and a sequence $\{L_k, k \geq 1\}$ of random variables i.i.d. with distribution $\mathrm{Q}(z, \cdot)$, we define iteratively, starting from (ξ_0, χ_0),

$$\begin{cases} \xi_{k+1} &= \xi_k - \gamma_{k+1} H_1(\xi_k, L_{k+1}) \\ \chi_{k+1} &= \chi_k - \gamma_{k+1} H_2(\xi_k, \chi_k, L_{k+1}). \end{cases} \tag{2.2.8}$$

Remark 2.2.1 *In the case where* $\gamma_k = \frac{1}{k}$*, the first line in* (2.2.8)*, for the value-at-risk specification of* H_1 *in* (2.2.3)*, is equivalent to*

$$\frac{1}{k} \sum_{l=1}^{k} \frac{\xi_{l-1}}{\xi_k} \mathbf{1}_{L_l > \xi_{l-1}} = 1 - \alpha,\ k \geq 1,$$

to be compared with the following empirical quantile ξ_k' *specification :*

$$\frac{1}{k} \sum_{l=1}^{k} \mathbf{1}_{L_l > \xi_k'} \approx 1 - \alpha.$$

The (almost-sure) limit $(\xi_\infty, \chi_\infty)$ of any convergent sequence $\{(\xi_k, \chi_k, k \geq 0\}$ is a solution to

$$(\xi, \chi) \mapsto \begin{cases} \mathbb{E}_0 \left[H_1(\xi, L)\right] = 0, \\ \mathbb{E}_0 \left[H_2(\xi, \chi, L)\right] = 0. \end{cases}$$

Therefore, any limit is a pair of solutions to (2.2.4)–(2.2.5). In particular, $\chi_\infty = \mathrm{ES}(z)$.

However, in our case, i.i.d. samples from the law of L are not available, because of the quantities $\mathbb{E}_1[\psi]$ and $\mathbb{E}_0[\psi']$ in L, which are not explicit. Therefore, we propose to replace exact sampling of L by approximate sampling. Toward this aim, we introduce two strategies.

Introducing i.i.d. $\{(\phi^k, \beta^k, Z_1^k), k \geq 1\}$ with the same distribution as (ϕ, β, Z_1) conditionally on $Z_0 = z$, the first strategy consists in replacing the draws $\{L_k, k \geq 1\}$ in (2.2.8) by

$$\phi_k + \frac{\beta_k}{M_k} \sum_{m=1}^{M_k} \psi_{m,k} - \frac{1}{M_k'} \sum_{m=1}^{M_k'} \psi_m', \tag{2.2.9}$$

where, conditionally on Z_1^k, $\{\psi_{m,k}, m \geq 1\}$ are i.i.d. with distribution $\Pi(Z_1^k, \cdot)$; conditionally on $Z_0 = z$, $\{\psi'_m, m \geq 1\}$ are i.i.d. with distribution $\Pi'(z, \cdot)$; M_k, M'_k are positive integers.

Of course, conditionally on $Z_0 = z$, the second average in (2.2.9) can be updated at each step k using only the corresponding partial sum at step $k-1$ and the samples $\{\psi'_m, M'_{k-1} < m \leq M'_k\}$.

The second strategy consists in replacing the draws $\{L_k, k \geq 1\}$ in (2.2.8) by

$$\phi_k + \beta_k \widehat{\Psi}(Z_1^k) - \frac{1}{M'_k} \sum_{m=1}^{M'_k} \psi'_m,$$

where the first and last terms are as before and where $\widehat{\Psi}(\cdot)$ is a regression-based estimator, computed prior and independently from the Z_1^k, of the function $\Psi(\cdot)$, such that

$$\Psi(Z_1) = \mathbb{E}_0 [\psi | Z_1], \qquad \mathrm{P}(z, \cdot)\text{-a.s} \tag{2.2.10}$$

(recall that $\mathrm{P}(z, \cdot)$ denotes the conditional distribution of Z_1 given $Z_0 = z$).

The advantage of the first approach is that, under sufficiently good convergence hypotheses for the nested averages (see the assumptions of Theorem 2.3.1), the approximation of $\mathrm{ES}(z)$ can be made asymptotically as good as desired. On the other side, the approach based on the regression requires a previous knowledge of the global behavior of Ψ (as an element of a certain function space) in order to give approximations with small bias (see Theorem 2.4.1), which is essential to have good asymptotics in our error analysis. Nevertheless, the second strategy has a small computational cost compared with the first one (at least for large values of the M_k in (2.2.9)). This can be a significant advantage if we indeed know which function space can serve to build a good predictor of the function Ψ.

Algorithmic summaries of these two strategies are given in the respective Sections 2.2.3 and 2.2.4. In Section 2.2.2, for pedagogical purposes, we start by recalling essentially known results in the base case where $\psi = \psi' = 0$.

2.2.2 Base-case Without Present and Future Liabilities

```
Input : A positive sequence $\{\gamma_k, k \geq 1\}$, $K \in \mathbb{N}^*$, $\xi_0 \in \mathbb{R}$, $\chi_0 \in \mathbb{R}$, and $z \in \mathbb{R}^q$.
for $k = 1$ to $K$, do
    /* Sampling step */
    Sample $\phi^k$ with the same distribution as $\phi$ conditionally on $Z_0 = z$, independently from the past draws ;
    Set $L_k := \phi^k$ ;
    /* Update the conditional VaR and ES estimates */
    $\xi_k = \xi_{k-1} - \gamma_k H_1(\xi_{k-1}, L_k)$ ;
    $\chi_k = \chi_{k-1} - \gamma_k H_2(\xi_{k-1}, \chi_{k-1}, L_k)$.
Return(The sequences $\{\chi_k, 1 \leq k \leq K\}$ and $\{\xi_k, 1 \leq k \leq K\}$)
```

Algorithm 1: Estimates of $\mathrm{VaR}(z)$ and $\mathrm{ES}(z)$ in the base case without present and future liabilities ($\psi = \psi' = 0$).

Note that, when $\psi \equiv \psi' = 0$, the random variables $\{L_k, k \geq 1\}$ are i.i.d. with distribution $\mathrm{Q}(z, \cdot)$. Therefore, sufficient conditions on this distribution and on the sequence $\{\gamma_k, k \geq 1\}$ for the almost-sure convergence of $\{\xi_k, k \geq 0\}$ to $\mathrm{VaR}(z)$ and $\{\chi_k, k \geq 0\}$ to $\mathrm{ES}(z)$ can be proven by application of standard results for stochastic approximation algorithms : By application of Theorem 2.7.1 and Lemma 2.7.2 in Appendix 2.7.2, we prove in Theorem 2.2.1 that the algorithm produces a sequence $\{(\xi_k, \chi_k), k \geq 1\}$ converging to a pair solution of (2.2.4)-(2.2.5) where $L \sim \mathrm{Q}(z, \cdot)$. Hence, ξ_K is a strongly consistent estimator of a value-at-risk of level α of the distribution $\mathrm{Q}(z, \cdot)$, while χ_K is a (strongly) consistent estimator of the associated expected shortfall.

More precisely, these convergences are established under the following assumptions.

H3 *$\{\gamma_k, k \geq 1\}$ is a $(0,1)$-valued deterministic sequence, such that for some $\kappa \in (0,1]$,*

$$\sum_k \gamma_k = +\infty, \qquad \sum_k \gamma_k^{1+\kappa} < +\infty.$$

H4 *a) Under $\mathbb{P}_0$, $L = \phi \sim \mathrm{Q}(Z_0 = z, \cdot)$ has a continuous cumulative distribution function.*

b) $\mathbb{E}_0\left[L^2\right] = \int x^2 \mathrm{Q}(z, \mathrm{d}x) < +\infty$.

H3 is standard in stochastic approximation, and is satisfied for example with $\gamma_n \sim \gamma_\star / n^c$ and $c \in (1/2, 1]$. The condition H4 essentially allows to characterize the set of the limiting points of the algorithm and to prove that the stochastic approximation algorithm is a perturbation of a discretized ODE with a controlled noise.

Theorem 2.2.1 *Let $\{(\xi_k, \chi_k), k \geq 1\}$ be the output of Algorithm 1. Assume H3 and H4. Then there exist a bounded random variable ξ_∞ and a real number χ_∞ satisfying (2.2.6) $\mathbb{P}_0$-a.s. and such that, for any $p \in (0,2)$,*

$$\mathbb{P}_0\left(\lim_{k\to\infty} (\xi_k, \chi_k) = (\xi_\infty, \chi_\infty)\right) = 1, \qquad \lim_{k\to\infty} \mathbb{E}_0\left[|\xi_k - \xi_\infty|^p\right] = 0.$$

The proof of this result, which is very close to [28, Theorem 1], is detailed in Appendix 2.7.3. The proof consists in first proving the almost-sure convergence of the sequence $\{\xi_k, k \geq 0\}$ toward the set of solutions of (2.2.4) by applying classical results on the convergence of stochastic approximation scheme ; for the sake of completeness these results are stated and proved as Theorem 2.7.1 in Appendix 2.7.2. We then deduce the convergence of the sequence $\{\chi_k, k \geq 0\}$ by using the fact that χ_k can be written as a weighted sum of the samples $\{L_j, \xi_j, 0 \leq j \leq k\}$ (see Lemma 2.7.2 in Appendix 2.7.1).

Remember that although the set of solutions to the equation $\xi : 1 - \alpha = \int_\xi^\infty \mathrm{Q}(z, \mathrm{d}x)$ might not be a singleton (when $\mathrm{VaR}(z)$ is not unique), $\mathrm{ES}(z)$ is unique - see Lemma 2.7.1 in Appendix 2.7.3.

2.2.3 With Future Liability Estimated by Nested Monte Carlo

```
Input : A positive sequence $\{\gamma_k, k \geq 1\}$, $\mathbb{N}^*$-valued sequences
    $\{M_k, M'_k, k \geq 1\}$, $\xi_0 \in \mathbb{R}$, $\chi_0 \in \mathbb{R}$, $S'_0 = 0$, $M'_0 = 0$ and $z \in \mathbb{R}^q$.
for $k = 1$ to $K$, do
    /* Sampling step */
    Sample $(\phi^k, \beta^k, Z_1^k)$ with the same distribution as $(\phi, \beta, Z_1)$
      conditionally on $Z_0 = z$, independently from the past draws ;
    Sample $(M'_k - M'_{k-1})$ independent copies $\{\psi'_m, M'_{k-1} < m \leq M'_k\}$
      with the distribution $\Pi'(z, \cdot)$, independently from the past
      draws;
    Given $Z_1^k$, sample $M_k$ independent copies $\{\psi_{m,k}, 1 \leq m \leq M_k\}$
      with the distribution $\Pi(Z_1^k, \cdot)$;
    Compute
    $S'_k := S'_{k-1} + \sum_{m=M'_{k-1}+1}^{M'_k} \psi'_m$;
    $L_k := \phi^k + \beta^k \frac{1}{M_k} \sum_{m=1}^{M_k} \psi_{m,k} - \frac{1}{M'_k} S'_k$ ;
    /* Update the conditional VaR and ES estimates */
    $\xi_k = \xi_{k-1} - \gamma_k H_1(\xi_{k-1}, L_k)$ ;
    $\chi_k = \chi_{k-1} - \gamma_k H_2(\xi_{k-1}, \chi_{k-1}, L_k)$.
Return(The sequences $\{\chi_k, 1 \leq k \leq K\}$ and $\{\xi_k, 1 \leq k \leq K\}$)
```

Algorithm 2: Estimates of VaR(z) and ES(z) with future liability estimated by nested Monte Carlo.

Note that the random variables $\{L_k, k \geq 1\}$ have the same distribution, but this distribution is not $\mathrm{Q}(z, \cdot)$, the distribution of L given by (2.2.1) : there is a bias which, roughly speaking, can be made as small as possible by choosing M_k, M'_k large enough.

We provide in Section 2.3.1 sufficient conditions on $\mathrm{Q}(z, \cdot)$ and on the sequences $\{\gamma_k, k \geq 1\}$, $\{M_k, k \geq 1\}$, $\{M'_k, k \geq 1\}$ for the $\mathbb{P}_0$-a.s. convergence of $\{\xi_k, k \geq 0\}$ to VaR(z) and $\{\chi_k, k \geq 0\}$ to ES(z). We also provide convergence rates in Section 2.3.2 and show the benefit of considering the averaged outputs $K^{-1} \sum_{k=1}^K \xi_k$ and $K^{-1} \sum_{k=1}^K \chi_k$ as estimators of VaR(z) and ES(z).

2.2.4 With Future Liability Estimated by Regression

The regression approach relies on the following observation : The function Ψ in (2.2.10) satisfies

$$\Psi = \mathrm{argmin}_{h \in \mathcal{H}} \int_{\mathbb{R}^q} \int_{\mathbb{R}} (w - h(z_1))^2 \, \Pi(z_1, \mathrm{d}w) \mathrm{P}(z, \mathrm{d}z_1), \qquad (2.2.11)$$

where $\mathcal{H}$ denotes the set of Borel measurable, $\mathrm{P}(z, \cdot)$-square integrable functions from $\mathbb{R}^q$ to $\mathbb{R}$. Since the integral in (2.2.11) is not explicit but sampling from the conditional distribution of (ψ, Z_1) given $Z_0 = z$ is possible, we define the estimate $\widehat{\Psi}(\cdot)$ as the solution of the empirical criterion associated with (2.2.11), replacing this integral by a Monte Carlo sum with i.i.d. samples.

If furthermore we replace the 'complex' functional space $\mathcal{H}$ by a space $\widehat{\mathcal{H}}$ suitable to least squares estimation, typically a finite-dimensional vector space of functions (but not necessarily, possibly also e.g. a neural network), we obtain a version of

(2.2.11) in which Ψ is approximated by the solution to a least squares regression problem. The best choice for $\widehat{\mathcal{H}}$ will depend on the specific problem at hand, typically on regularity assumptions regarding Ψ.

In order to make use of the distribution-free theory of non-parametric regression (as explained, for instance, in [84]), it is better to deal with bounded random variables to get nice statistical error estimates (through appropriate measure concentration inequalities). For this reason we consider the projection of the real-valued random variable ψ on the interval $[-B, B]$:

$$\psi^B := \psi \mathbf{1}_{[|\psi| \leq B]} + B \ \mathrm{sign}(\psi) \ \mathbf{1}_{[|\psi| > B]}, \tag{2.2.12}$$

where B is a large threshold assumed to be known by the user, and we write $\Psi^B(Z_1) := \mathbb{E}_0[\psi^B | Z_1]$.

This gives rise to the following Algorithm 3 for the estimation of $\mathrm{ES}(z)$, using the embedded regression Algorithm 4.

```
Input : $B > 0$, $M, M' \in \mathbb{N}^*$, $K \in \mathbb{N}^*$, a finite dimensional vector space
    $\widehat{\mathcal{H}}$ of Borel measurable functions from $\mathbb{R}^q$ to $\mathbb{R}$, a positive
    sequence $\{\gamma_k, k \geq 1\}$, $\xi_0 \in \mathbb{R}$, $\chi_0 \in \mathbb{R}$ and $z \in \mathbb{R}^q$.
/* Regression step                                                    */
Compute an approximation $\widehat{\Psi^B}(\cdot)$ of $\Psi(\cdot)$ by Algorithm 4 with
    inputs $B, M$ and $\widehat{\mathcal{H}}$.
/* stochastic approximation step                                      */
Sample $M'$ independent copies $\{\psi'_m, 1 \leq m \leq M'\}$ with the
    distribution $\Pi'(z, \cdot)$;
for $k = 1$ to $K$, do
    Sample $(\phi^k, \beta^k, Z_1^k)$ from the conditional distribution of $(\phi, \beta, Z_1)$
      given $Z_0 = z$, independently from the past draws; Compute
      $L_k := \phi^k + \beta^k \widehat{\Psi^B}(Z_1^k) - \frac{1}{M'} \sum_{m=1}^{M'} \psi'_m$ ;
    /* Update the conditional VaR and ES estimates                    */
    $\xi_k = \xi_{k-1} - \gamma_k H_1(\xi_{k-1}, L_k)$ ;
    $\chi_k = \chi_{k-1} - \gamma_k H_2(\xi_{k-1}, \chi_{k-1}, L_k)$.
Return(The sequences $\{\chi_k, 1 \leq k \leq K\}$ and $\{\xi_k, 1 \leq k \leq K\}$)
```

Algorithm 3: Estimates of $\mathrm{VaR}(z)$ and $\mathrm{ES}(z)$ with future liability estimated by regression.

```
Input : $B > 0$, $M \in \mathbb{N}^*$, a function space $\widehat{\mathcal{H}}$ of Borel measurable
    functions from $\mathbb{R}^q$ to $\mathbb{R}$,
Sample $M$ independent copies $\mathcal{D} = \{(\psi_m, Z_1^m), m = 1, \cdots, M\}$ from
    the conditional distribution of $(\psi, Z_1)$ given $Z_0 = z$ ;
Compute
    $\widetilde{h} := \arg\min_{h \in \widehat{\mathcal{H}}} \frac{1}{M} \sum_{m=1}^{M} \left(\psi_m^B - h(Z_1^m)\right)^2$
  and set
    $\widehat{\Psi^B}(\cdot) := \mathrm{sign}(\widetilde{h}(\cdot)) \left(|\widetilde{h}(\cdot)| \wedge B\right)$ ;
  Return(The function $\widehat{\Psi^B}(\cdot)$)
```

Algorithm 4: Approximation of Ψ in (2.2.10) by empirical regression.

The analysis of Algorithm 3 is established in Section 2.4. Theorem 2.4.1 gives a control of the deviation in the $\mathrm{L}^1_{\mathbb{P}_0}$−norm of the respective limits $|\xi_\infty - \mathrm{VaR}(z)|$ and $|\chi_\infty - \mathrm{ES}(z)]$, where $(\xi_\infty, \chi_\infty)$ is the (almost-sure) limit of (ξ_K, χ_K) as K goes to infinity, from biases (or"deterministic errors") given, up to multiplicative constants, as the respective square and cube roots of

$$\inf_{h \in \widehat{\mathcal{H}}} \mathbb{E}_0|h(Z_1) - \Psi^B(Z_1)|^2 + \mathbb{E}_0[((|\psi| - B)^+)^2]. \tag{2.2.13}$$

The control of (2.2.13) depends on analytic features of the problem at hand, typically on the regularity of Ψ for the choice of $\widehat{\mathcal{H}}$ and on the distribution of ψ for the choice of B (see [84, Chapter 10] for a general discussion). [2] Putting everything together, these results are telling us how we should choose the inputs $(\widehat{\mathcal{H}}, B, M, M')$ in order to make the limit $(\chi_\infty, \xi_\infty)$ of the (χ_K, ξ_K) as close as desired from the target values $(\chi_\star, \xi_\star)$.

2.3 Convergence Analysis of the Economic Capital SA Algorithm 2 (Future Liabilities Estimated by Nested Monte Carlo)

Section 2.3.1 deals with the almost-sure convergence of Algorithm 2. Section 2.3.2 addresses the rate of convergence of Algorithm 2 along a converging sequence : a central limit theorem is established as well as the rate of convergence when an averaging technique is applied to the output of Algorithm 2.

2.3.1 Almost-sure Convergence

The difference between Algorithm 1 and Algorithm 2 is that $\beta\mathbb{E}_0\left[\psi|Z_1\right] - \mathbb{E}_0\left[\psi'|Z_0\right]$ in the definition of L (see (2.2.1)) is non zero. The expectations are untractable and they are approximated by Monte Carlo sums. Hence, the random variables $\{L_k, k \geq 1\}$ in Algorithm 2 are no more i.i.d. under the distribution $\mathrm{Q}(z, \cdot)$. Nevertheless, when the number of Monte Carlo points tends to infinity, the Monte Carlo error vanishes, and it is expected that Algorithm 2 inherits the same asymptotic behavior as the one of Algorithm 1, in which the L_k are i.i.d. with distribution $\mathrm{Q}(z, \cdot)$. We provide sufficient conditions for this intuition to hold. H7 strenghtens H3 by showing how the stepsize γ_k and the number of Monte Carlo points M_k, M'_k have to be balanced ; H5 is in echo to H4. H6 (see also H8) is introduced to control the bias between the distributions of the L_k and $\mathrm{Q}(z, \cdot)$.

We assume

H5 *Under $\mathbb{P}_0$, $L := \phi + \beta\mathbb{E}_0\left[\psi|Z_1\right] - \mathbb{E}_0\left[\psi'|Z_0 = z\right] \sim \mathrm{Q}(z, \cdot)$ and it has a density with respect to the Lebesgue measure on $\mathbb{R}$, bounded by $C_0(z) > 0$. In addition,*

$$\mathbb{E}_0\left[|L|^2\right] = \int x^2 \mathrm{Q}(z, \mathrm{d}x) < +\infty.$$

[2] Note also that the term $\mathbb{E}_0[((|\psi| - B)^+)^2]$ controls the truncation error $\mathbb{E}_0|\Psi(Z_1) - \Psi^B(Z_1)|^2$ by Jensen's inequality :

$$\mathbb{E}_0|\Psi(Z_1) - \Psi^B(Z_1)|^2 = \mathbb{E}_0|\mathbb{E}_0[(\psi - \psi^B)|Z_1]|^2 \leq \mathbb{E}_0|\psi - \psi^B|^2 = \mathbb{E}_0[((|\psi| - B)^+)^2].$$

H6 *There exists $p_\star \geq 2$ such that*

$$C_{p_\star}(z) := \mathbb{E}_0\left[\int \left|w - \int u\Pi(Z_1, \mathrm{d}u)\right|^{p_\star} \Pi(Z_1, \mathrm{d}w)\right] + \int \left|w - \int u\Pi'(z, \mathrm{d}u)\right|^{p_\star} \Pi'(z, \mathrm{d}w)$$

is finite.

H7 *The sequences $\{M_k, k \geq 1\}$ and $\{M'_k, k \geq 1\}$ are $\mathbb{N}^\star$-valued, $\{\gamma_k, k \geq 1\}$ is a $(0,1)$-valued sequence, and there exists $\kappa \in (0,1]$ such that*

$$\sum_k \gamma_k = +\infty, \qquad \sum_k \gamma_k^{1+\kappa} < +\infty, \tag{2.3.1}$$

$$\sum_{k\geq 1} \gamma_k^{1-\kappa}\, (M_k \wedge M'_k)^{-p_\star/(1+p_\star)} < +\infty, \qquad \sum_{k\geq 1} \gamma_k\, (M_k \wedge M'_k)^{-1/2} < +\infty. \tag{2.3.2}$$

Let us discuss the condition H7 in the case $\gamma_k \sim \gamma_\star k^{-c}(\ln k)^{-\bar{c}}$ and $(M_k \wedge M'_k) \sim m_\star k^{\mu}(\ln k)^{\bar{\mu}}$ when $k \to +\infty$ (for some $c, \mu \geq 0$). Then (2.3.1) in H7 implies that $c \in [1/2, 1]$ (the case $c = 1/2$ implies $\kappa = 1$ and $\bar{c} > 1$).

When $c = 1$, we have to choose $\bar{c} \leq 1$ and $\mu > 0$ (note that the last condition in (2.3.2) does not allow $\mu = 0$). Therefore, the number of Monte Carlo points has to increase, even slowly, along the iterations ; this comes from the fact that the Monte Carlo bias has to vanish along iterations to force Algorithm 2 to have the same behavior as Algorithm 1.

When $c = 1/2$, the slowest rate for $M_k \wedge M'_k$ is $\mu = 1 + 1/p_\star$, and in that case, $\bar{\mu} > 1 + 1/p_\star$ and $\bar{c} > 1$. Therefore, the number of Monte Carlo points has to increase more than linearly with k.

When $c \in (1/2, 1)$, the slowest rate for $M_k \wedge M'_k$ is $\mu = 2(1-c)(1 + 1/p_\star)$, and in that case, $\bar{c} > 1/c$ and $\bar{\mu} > (1 + 1/p_\star)(1 - \bar{c}(2 - 1/c))$.

The above discussion makes it apparent that either we choose a rapidly decaying stepsize sequence, and we have the weakest Monte Carlo cost ; or we choose a slowly decaying stepsize sequence, but the number of Monte Carlo points has to increase more than linearly. It is known that for implementation efficiency, a slow decaying rate for γ_k is better during the burn-in phase of the algorithm (while it has not reached its asymptotic convergence rate).

If H6 is strenghtened into

H8 *There exists $C_\infty(z) > 0$ such that for any $\delta > 0$ and any integer M,*

$$\mathbb{P}_0\left(\left|\frac{1}{M}\sum_{m=1}^{M} \psi_m - \mathbb{E}_0\left[\psi | Z_1\right]\right| > \delta\right) \vee \mathbb{P}_0\left(\left|\frac{1}{M}\sum_{m=1}^{M} \psi'_m - \mathbb{E}_0\left[\psi' | Z_0\right]\right| > \delta\right) \leq e^{-C_\infty(z)\, M\, \delta^2},$$

where conditionally to (Z_0, Z_1), $\{\psi_m, m \geq 1\}$ are i.i.d. with distribution $\Pi(Z_1, \mathrm{d}w)$, and $\{\psi'_m, m \geq 1\}$ are i.i.d. with distribution $\Pi'(z, \mathrm{d}w)$.

then the condition (2.3.2) in H7 is weakened into

$$\sum_{k\geq 1} \gamma_k^{1-\kappa} \frac{\ln(M_k \wedge M'_k)}{(M_k \wedge M'_k)} < +\infty, \qquad \sum_{k\geq 1} \gamma_k\, (M_k \wedge M'_k)^{-1/2} < +\infty.$$

The above discussion on the choice of (c, μ) is essentially modified as follows (the choice of the logarithmic terms $\bar{c}, \bar{\mu}$ is not detailed) : either $c = 1$ and $\mu > 0$, or $c \in [1/2, 1)$ and $\mu = 2(1-c)$.

The following proposition is fundamental in the proof of Theorem 2.3.1. It allows to control the error induced by drawing samples L_k under a distribution approximating $\mathrm{Q}(z,\cdot)$ instead of sampling from $\mathrm{Q}(z,\cdot)$. Its proof is postponed to Appendix 2.7.4.

Lemma 2.3.1 *Assume H5 and H6. Let L' be a random variable such that*

$$|L-L'| \leq c_\beta \left|\frac{1}{M}\sum_{m=1}^{M}\psi_m - \int w\Pi(z_1,\mathrm{d}w)\right| + \left|\frac{1}{M'}\sum_{m=1}^{M'}\psi'_m - \int w\Pi'(z,\mathrm{d}w)\right|,$$

where conditionally on (Z_1,Z_0), $\{\psi_m, m\geq 1\}$ (resp. $\{\psi'_m, m\geq 1\}$) are i.i.d. with distribution $\Pi(Z_1,\cdot)$ (resp. $\Pi'(z,\cdot)$). Then,

$$\sup_{\xi\in\mathbb{R}} \mathbb{E}_0\left[|\mathbf{1}_{L>\xi}-\mathbf{1}_{L'>\xi}|\right] \leq (1\vee c_\beta)^{p_\star}\ \frac{2^{p_\star}\ (C_0(z)+c_{p_\star}C_{p_\star}(z))}{(M\wedge M')^{p_\star/(2(1+p_\star))}}, \tag{2.3.3}$$

$$\sup_{\xi\in\mathbb{R}} \mathbb{E}_0\left[\left|(L-\xi)^+-(L'-\xi)^+\right|^{p_\star}\right] \leq (1\vee c_\beta)^{p_\star}\ \frac{c_{p_\star}C_{p_\star}(z)}{(M\wedge M')^{p_\star/2}}, \tag{2.3.4}$$

where $c_{p_\star}$ is a universal constant depending only on $p_\star$. When H6 is replaced with H8, then for any $M, M'\geq 3$,

$$\sup_{\xi\in\mathbb{R}} \mathbb{E}_0\left[|\mathbf{1}_{L>\xi}-\mathbf{1}_{L'>\xi}|\right] \leq 2\left(1+\frac{C_0(z)}{\sqrt{2\,C_\infty(z)}}\right)\sqrt{\frac{\ln(M\wedge M')}{(M\wedge M')}}. \tag{2.3.5}$$

We can now prove that the output of Algorithm 2 provides strongly consistent estimators of $\mathrm{VaR}(z)$ and $\mathrm{ES}(z)$. The proof of the next theorem is postponed to Appendix 2.7.4.

Theorem 2.3.1 *Let $\{(\xi_k,\chi_k), k\geq 1\}$ be the output of Algorithm 2. Assume H5, H6, and H7. Then there exists a bounded random variable ξ_∞ and a real χ_∞ satisfying $\mathbb{P}_0$-a.s. (2.2.6) and such that for any $p\in(0,2)$*

$$\mathbb{P}_0\left(\lim_{k\to\infty}(\xi_k,\chi_k)=(\xi_\infty,\chi_\infty)\right)=1, \qquad \lim_{k\to\infty}\mathbb{E}_0\left[|\xi_k-\xi_\infty|^p\right]=0.$$

2.3.2 Rates of Convergence of Algorithm 2

We establish a rate of convergence in L^2 and a central limit theorem, along a sequence $\{(\xi_k,\chi_k), k\geq 1\}$ converging to $(\xi_\star,\chi_\star)$, where $(\xi_\star,\chi_\star)$ is a solution to (2.2.6) ; this solution is fixed throughout this section. These results are derived under the following conditions.

H 9 *$(\xi_\star,\chi_\star)$ solves (2.2.6). H5 holds and is strenghtened as follows : under $\mathbb{P}_0$, the density of $L:=\phi+\beta\mathbb{E}_0\left[\psi|Z_1\right]-\mathbb{E}_0\left[\psi'|Z_0=z\right]\sim \mathrm{Q}(z,\cdot)$ w.r.t. the Lebesgue measure on $\mathbb{R}$, denoted by $f(z,\cdot)$, is continuously differentiable in a neighborhood of $\xi_\star$ and strictly positive at $\xi_\star$. In addition, there exists $\nu_\star>0$ such that*

$$\mathbb{E}_0\left[|L|^{2+\nu_\star}\right]=\int |x|^{2+\nu_\star}\mathrm{Q}(z,\cdot)<+\infty.$$

H6 is strenghtened as follows : there exists $p_\star>2$ such that

$$C_{p_\star}(z):=\mathbb{E}_0\left[\int\left|w-\int u\Pi(Z_1,\mathrm{d}u)\right|^{p_\star}\Pi(Z_1,\mathrm{d}w)\right]+\int\left|w-\int u\Pi'(z,\mathrm{d}u)\right|^{p_\star}\Pi'(z,\mathrm{d}w) \tag{2.3.6}$$

is finite.

To make the assumptions simpler, we consider the case where the stepsize sequence $\{\gamma_k, k \geq 1\}$ is polynomially decreasing.

H10 *When $k \to \infty$, $\gamma_k \sim \gamma_\star k^{-c}$ where $c \in (1/2, 1]$ and $\gamma_\star > 0$; in the case $c = 1$, $2\gamma_\star > (1 \wedge (f(z, \xi_\star)/(1-\alpha)))^{-1}$. In addition, c, M_k, M'_k satisfy*

$$\lim_k k^{c/2} \left(M_k \wedge M'_k\right)^{-p_\star/(2(1+p_\star))} = 0, \tag{2.3.7}$$

where $p_\star$ is given by H9.

When $M_k \wedge M'_k \sim m_\star k^\mu$ when $k \to \infty$, the condition (2.3.7) is satisfied with $\mu > c(1 + 1/p_\star)$. In the case the condition (2.3.6) is replaced with H8, the condition (2.3.7) gets into

$$\lim_k k^{c/2} \sqrt{\frac{\ln(M_k \wedge M'_k)}{M_k \wedge M'_k}} = 0,$$

which is satisfied with $\mu > c$.

Set $\theta_k := (\xi_k, \chi_k)$ and $\theta_\star := (\xi_\star, \chi_\star)$. Lemma 2.3.2 shows that $\theta_k - \theta_\star$ is bounded, in some sense, by $\sqrt{\gamma_k}$. Theorem 2.3.2 provides a central limit theorem, proving that, along converging paths, the normalized error $\gamma_k^{-1/2}\ (\theta_k - \theta_\star)$ behaves asymptotically as a Gaussian distribution.

Lemma 2.3.2 *Assume H9 and H10. Then, there exist positive random variables X_k, Y_k such that $\mathbb{P}_0(\sup_k |X_k| < \infty) = 1$, $\sup_k \mathbb{E}_0\left[|Y_k|\right] < \infty$ and*

$$\gamma_k^{-1}\ |\theta_k - \theta_\star|^2\, \mathbf{1}_{\lim_q \theta_q = \theta_\star} \leq X_k\ Y_k.$$

Set

$$\Gamma_c := \frac{1}{2}\begin{bmatrix} \alpha(1-\alpha)^2 \left(f(z, \xi_\star)\right)^{-1} & 2\alpha\left(1 + \frac{f(z,\xi_\star)}{1-\alpha}\right)^{-1} \\ 2\alpha\left(1 + \frac{f(z,\xi_\star)}{1-\alpha}\right)^{-1} & \mathbb{V}\mathrm{ar}_0\left[(L - \xi_\star)^+\right] \end{bmatrix}$$

and

$$\Gamma_1 := \begin{bmatrix} (1-\alpha)\left(2\gamma_\star \frac{f(z,\xi_\star)}{1-\alpha} - 1\right)^{-1} & \mathbb{E}_0\left[(L-\xi_\star)^+\right]\left(\gamma_\star(1 + \frac{f(z,\xi_\star)}{1-\alpha}) - 1\right)^{-1} \\ \mathbb{E}_0\left[(L-\xi_\star)^+\right]\left(\gamma_\star(1 + \frac{f(z,\xi_\star)}{1-\alpha}) - 1\right)^{-1} & \mathbb{V}\mathrm{ar}_0\left[(L-\xi_\star)^+\right](2\gamma_\star - 1)^{-1}(\alpha\gamma_\star)^{-1}. \end{bmatrix}$$

Theorem 2.3.2 *Assume H9 and H10. Let $\{\theta_k, k \geq 1\}$ be the output of Algorithm 2. Then, under the conditional probability $\mathbb{P}_0\left(\cdot | \lim_q \theta_q = \theta_\star\right)$, the sequence $\{\gamma_k^{-1/2}(\theta_k - \theta_\star), k \geq 1\}$ converges in distribution to the centered bivariate normal distribution with covariance matrix $(1-\alpha)^{-2}\Gamma_c$ in the case $c \in (1/2, 1)$, and $(1-\alpha)^{-2}\alpha\gamma_\star\Gamma_1$ in the case $c = 1$ (where c is given by H10).*

The proof of Theorem 2.3.2 is postponed to Appendix 2.7.6. Lemma 2.3.2 is a consequence of [57, Lemma 3.1.], applied to the same decomposition of $\theta_k - \theta_\star$ as in the proof of Theorem 2.3.2 ; details are omitted.

Theorem 2.3.2 shows that *(i)* the maximal rate of convergence is reached with a stepsize γ_k decaying at a rate $1/k$ as soon as $\gamma_\star$ is large enough (see H10 in the case $c = 1$) ; *(ii)* the limiting variance depends on $\gamma_\star$. In practice, the condition on $\gamma_\star$ is difficult to check since the quantity $f(z, \xi_\star)$ is unknown in many applications ;

in addition, it is known (see e.g. [32, Lemma 4, Chapter 3, Part I] or [57, Section 3]) that the optimal variance for an SA algorithm targeting the roots of the function

$$\theta = (\xi, \chi) \mapsto \begin{bmatrix} 1 - (1-\alpha)^{-1}\mathbb{P}_0(L > \xi) \\ \chi - \xi - (1-\alpha)^{-1}\mathbb{E}_0\left[(L-\xi)^+\right] \end{bmatrix}$$

is given by

$$\Gamma_\star := \begin{bmatrix} \frac{\alpha(1-\alpha)}{f^2(z,\xi_\star)} & \frac{\alpha}{1-\alpha}\frac{\mathbb{E}_0\left[(L-\xi_\star)^+\right]}{f(z,\xi_\star)} \\ \frac{\alpha}{1-\alpha}\frac{\mathbb{E}_0\left[(L-\xi_\star)^+\right]}{f(z,\xi_\star)} & \frac{\mathbb{V}\mathrm{ar}_0\left[(L-\xi_\star)^+\right]}{(1-\alpha)^2} \end{bmatrix}.$$

We prove in Theorem 2.3.3 that the optimal rate $O(1/k)$ and this optimal limiting variance $\Gamma_\star$ can be obtained by a simple post-processing of the output of Algorithm 2 run with $\gamma_k \sim \gamma_\star k^{-c}$ for some $c \in (1/2, 1)$. The proof of Theorem 2.3.3 is postponed to Appendix 2.7.6. This post-processing technique is known in the literature as the Polyak-Ruppert averaging (see [101, 102]). Set

$$\bar{\theta}_k := \frac{1}{k}\sum_{\ell=1}^{k} \begin{bmatrix} \xi_\ell \\ \chi_\ell \end{bmatrix}$$

Theorem 2.3.3 *Let $\{\theta_k, k \geq 1\}$ be the output of Algorithm 2. Assume H9, $\gamma_k \sim \gamma_\star k^{-c}$ with $c \in (1/2, 1)$ and $\gamma_\star > 0$, and*

$$\lim_k k^c \left(M_k \wedge M'_k\right)^{-p_\star/(2(1+p_\star))} = 0, \qquad \lim_k k^{-1/2} \sum_{l=1}^{k} \left(M_l \wedge M'_l\right)^{-p_\star/(2(1+p_\star))} = 0. \tag{2.3.8}$$

Then, under the conditional probability $\mathbb{P}_0\left(\cdot \middle| \lim_q \theta_q = \theta_\star\right)$, the sequence $\{k^{1/2}\left(\bar{\theta}_k - \theta_\star\right), k \geq 1\}$ converges in distribution to the centered bivariate normal distribution with covariance matrix $\Gamma_\star$.

When $M_k \wedge M'_k \sim m_\star k^\mu$, (2.3.8) is satisfied with $\mu > 2c(1 + 1/p_\star)$. In the case where the condition (2.3.6) is replaced with H8 in Theorem 2.3.3, then the condition (2.3.8) becomes

$$\lim_k k^c \frac{\ln(M_k \wedge M'_k)}{(M_k \wedge M'_k)^{1/2}} = 0, \qquad k^{-1/2}\sum_{l=1}^{k} \ln(M_l \wedge M'_l)\ (M_l \wedge M'_l)^{-1/2} = 0;$$

it is satisfied if $\mu > 2c$. Note that these conditions on μ are slightly more restrictive than what we obtained for the convergence of the sequence $\{\theta_k, k \geq 1\}$ in the case $c \in (1/2, 1)$.

2.4 Convergence Analysis of the Economic Capital SA Algorithm 3 (Future Liabilities Estimated by Regression)

In order to properly define $\widehat{\Psi^B}(Z_1)$ in $\widehat{L^B}$ as a random variable, we assume that the function space $\widehat{\mathcal{H}}$ is pointwise measurable.[3] We introduce the following object

[3] I.e. there exists a countable subfamily of $\widehat{\mathcal{H}}$ with the property that every function in $\widehat{\mathcal{H}}$ is a pointwise limit of these functions. It includes finite dimensional vector spaces, neural networks with continuous activation function,. . .

(cf. (2.4.1)) :

$$\Psi^B := \mathbb{E}_0[\psi^B | Z_1], \qquad L^B := \phi + \beta \Psi^B(Z_1) - \Psi'(z). \tag{2.4.1}$$

For any fixed $g \in \widehat{\mathcal{H}}$, we define

$$L_{g^B} := \phi + \beta\, g^B(Z_1) - \frac{1}{M'} \sum_{m=1}^{M'} \psi'_m, \tag{2.4.2}$$

where $g^B : \mathbb{R}^q \to \mathbb{R}$ is the truncation of g by B : $g^B := \mathrm{sign}(g)(|g| \wedge B)$. Last, for the approximation of Ψ^B obtained by regression (see Algorithm 4), we write

$$\widehat{L^B} := \phi + \beta\, \widehat{\Psi^B}(Z_1) - \frac{1}{M'} \sum_{m=1}^{M'} \psi'_m. \tag{2.4.3}$$

2.4.1 Existence of a Limit

H11 *ϕ, β, Z_1 are independent from the regression sample $\mathcal{D}$ (defined in Algorithm 4) and $\{\psi'_m : 1 \leq m \leq M'\}$ are i.i.d. with distribution $\mathbb{P}_0$. They are independent from the ψ_m. In addition we have the square integrability conditions : $\mathbb{E}_0\left[|\phi|^2 + |\psi'|^2\right] < +\infty$.*

Observe that the above assumption ensures in particular that, for any $g \in \widehat{\mathcal{H}}$,

$$\mathbb{E}_0\left[|L_{g^B}|^2\right] < +\infty.$$

We require an additional condition on L_{g^B}.

H12 *For every $g \in \widehat{\mathcal{H}}$, L_{g^B} in (2.4.2) has a continuous cumulative distribution function under $\mathbb{P}_0$.*

Lemma 2.4.1 *Assume H3, H11, and H12. Let $\{(\xi_k, \chi_k), k \geq 1\}$ be the ouput of Algorithm 3. Then, conditionally on $\mathcal{D}$, there exist random variables $(\xi_\infty, \chi_\infty)$, finite a.s. which are a solution of (2.2.4)-(2.2.5) for $L = \widehat{L^B}$ under $\mathbb{P}_0(\cdot \mid \mathcal{D})$, and such that $\mathbb{P}_0\left(\lim_k(\xi_k, \chi_k) = (\xi_\infty, \chi_\infty) \mid \mathcal{D}\right) = 1$.*

Proof *Given $g \in \widehat{\mathcal{H}}$, H3, H11 and H12 imply that the hypotheses of Theorem 2.2.1 are verified for every L_{g^B} as in (2.4.2) under the distribution $\mathbb{P}_0$. Hence, for fixed $\mathcal{D}$, the same is true for $\widehat{L^B}$ in (2.4.3) under the conditional distribution $\mathbb{P}_0(\cdot \mid \mathcal{D})$. The conclusion follows by application of Theorem 2.2.1.*

2.4.2 Error Analysis With a Given Approximate Model for the Regression Function

The next step is to bound the error between the initial model for L and the truncated and approximate model L_{g^B}, where we use the function g^B (for a given $g \in \widehat{\mathcal{H}}$) as a model of Ψ. For this we need Assumption 4 a) on the cumulative distribution function of L in (2.2.2) and its stronger version

Assumption 2.4.1 *Assume Assumption H4 a). Denote by* $(\xi_\star, \chi_\star)$ *a solution to* (2.2.4)-(2.2.5) *with* $L = \phi + \beta\Psi(Z_1) - \Psi'(z)$. *The distribution of* L, *with* $\mathbb{P}_0$ *c.d.f.* F, *admits a density* f *under* $\mathbb{P}_0$ *bounded by* C_f, *this density is positive and continuous on a neighborhood of the interval*

$$[\xi_\star - \zeta, \xi_\star + \zeta], \tag{2.4.4}$$

where

$$\zeta := 2^{1/3}(2C_f + 1)\left(\mathbb{E}_0|\beta g^B(Z_1) - \beta\Psi(Z_1)|^2 + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3}.$$

Lemma 2.4.2 *Assume H11-H12 and H4 a), let* $g \in \widehat{\mathcal{H}}$ *be given and let* $(\xi_{g^B,\star}, \chi_{g^B,\star})$ *be a solution to* (2.2.4)-(2.2.5) *for* L *there defined by* L_{g^B} *in* (2.4.2), *then*

$$|\chi_{g^B,\star} - \chi_\star| \le \frac{1}{1-\alpha}\left(\mathbb{E}_0|\beta g^B(Z_1) - \beta\Psi(Z_1)| + \left(\frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/2}\right). \tag{2.4.5}$$

If the stronger condition H2.4.1 holds (for g), then

$$|\xi_{g^B,\star} - \xi_\star| \le 2^{1/3}\,(2C_f + 1) \sup_{x\in[\xi_\star-\zeta,\xi_\star+\zeta]} |f(F^{-1}(x))|^{-1}\left(\mathbb{E}_0|\beta g^B(Z_1) - \beta\Psi(Z_1)|^2 + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3}. \tag{2.4.6}$$

Proof *We begin by proving* (2.4.6), *by an application of Corollary 2.7.1. For this, we first estimate the Kolmogorov distance* $d_{\mathrm{kol}}(L_{g^B}, L)$ *: actually Corollary 2.7.2 with* $p = 2$ *gives*

$$d_{\mathrm{kol}}(L_{g^B}, L) \le (2C_f + 1)(\mathbb{E}_0|L_{g^B} - L|^2)^{1/3}. \tag{2.4.7}$$

The difference in the expectation (2.4.7) *is bounded as (see definitions* (2.2.2) *and* (2.4.2)*)*

$$|L_{g^B} - L|^2 \le 2\left|\beta g^B(Z_1) - \beta\Psi(Z_1)\right|^2 + 2\left(\frac{1}{M'}\sum_{m=1}^{M'}(\psi_m - \Psi'(z))\right)^2.$$

Therefore, we deduce

$$d_{\mathrm{kol}}(L_{g^B}, L) \le 2^{1/3}(2C_f + 1)\left(\mathbb{E}_0|\beta g^B(Z_1) - \beta\Psi(Z_1)|^2 + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3} = \zeta. \tag{2.4.8}$$

Consequently, we can apply Corollary 2.7.1 with $r = s = \zeta$, *to get* (2.4.6).

The inequality (2.4.5) *follows in an easier way via* (2.7.20) *in Lemma 2.7.6.*

2.4.3 Error Analysis for the Randomly Optimal Regression Function

Observe that by taking formally $g^B = \widehat{\Psi^B}$, we obtain, as a corollary of the previous proposition, a pathwise control between $(\xi_\infty, \chi_\infty)$ (associated to $\widehat{L^B}$) and $(\xi_\star, \chi_\star)$ (associated to L), for a given regression sample $\mathcal{D}$. By reintegrating over the learning sample $\mathcal{D}$, we shall obtain an estimate about the corresponding mean L_1

error. This strategy works nicely, in particular if we allow Assumption H2.4.1 to be valid with a

$$\zeta = 2^{1/3}(2C_f + 1)\left(\mathbb{E}_0|\beta g^B(Z_1) - \beta\Psi(Z_1)|^2 + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3}$$

uniform in the learning sample $\mathcal{D}$. For this, set

$$\zeta_\infty := 2^{1/3}(2C_f + 1)\left(C_\beta^2\mathbb{E}_0\Big[\max_{\varepsilon=\pm1}|\varepsilon B - \Psi(Z_1)|^2\Big] + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3} \tag{2.4.9}$$

which stands for a (rough) upper bound for ζ. This explains the following new assumption.

H 13 *Assume Assumption H2.4.1 with $\zeta = \zeta_\infty$ defined in* (2.4.9).

Regarding the error analysis about the limits of Algorithm 3 (given by Lemma 2.4.1), our main result is now the following.

Theorem 2.4.1 *Assume H3, H11, H12, and H13. Let $B > 0$ and let $\widehat{\mathcal{H}}$ be a pointwise measurable function space with finite Vapnik-Chervonenkis dimension*[4] $\mathbf{VC}_{\widehat{\mathcal{H}}}$. *Set*

$$\mathcal{E}(\widehat{\mathcal{H}}, M, B) := C_\star B^2\mathbf{VC}_{\widehat{\mathcal{H}}}\frac{(1+\ln(M))}{M} + 4\inf_{h\in\widehat{\mathcal{H}}}\mathbb{E}_0[|h(Z_1) - \Psi_B(Z_1)|^2] + 4\,\mathbb{E}_0[((|\psi| - B)^+)^2], \tag{2.4.10}$$

where $C_\star$ is the constant that appears in (2.7.26). *We have*

$$\mathbb{E}_0|\chi_\infty - \chi_\star| \le \frac{1}{1-\alpha}\left(C_\beta(\mathcal{E}(\widehat{\mathcal{H}}, M, B))^{1/2} + \Big(\frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\Big)^{1/2}\right) \tag{2.4.11}$$

$$\mathbb{E}_0|\xi_\infty - \xi_\star| \le 2^{1/3}\,(2C_f + 1)\sup_{x\in[\xi_\star-\zeta_\infty,\xi_\star+\zeta_\infty]}|f(F^{-1}(x))|^{-1}\left(|\beta|_\infty^2\mathcal{E}(\widehat{\mathcal{H}}, M, B) + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3}. \tag{2.4.12}$$

Theorem 2.4.1 gives a precise and useful guide for tuning the parameters all together. Namely, to make the (asymptotic) errors $\mathbb{E}_0|\chi_\infty - \chi_\star|$ and $\mathbb{E}_0|\xi_\infty - \xi_\star|$ less than some tolerance ϵ, we can choose $\widehat{\mathcal{H}}$ and B such that the “bias” given by the second line in (2.4.10) is sufficiently small ; then one can choose M and M' large enough so that the right hand sides in (2.4.11) are less than ϵ. Unsurprisingly, when the complexity of $\widehat{\mathcal{H}}$ increases, the bias term ($\inf_{h\in\widehat{\mathcal{H}}}\dots$) goes to 0 and the variance term explodes ($\mathbf{VC}_{\widehat{\mathcal{H}}} \to +\infty$), hence one has to find a trade-off between those types of error. When one increases the threshold B, the bias decreases $\mathbb{E}_0[((|\psi| - B)^+)^2]$ but the variance increases (factor $C_\star B^2\dots$).

Proof *First, by H3, H11, H12 and Lemma 2.4.1, the limits*

$$\xi_\infty = \lim_k \xi_k, \quad \chi_\infty = \lim_k \chi_k$$

[4] See [84, Section 9.4].

indeed exist for every fixed $\mathcal{D}$ and they correspond to solutions of (2.2.4)-(2.2.5) *for $L = \widehat{L^B}$ (see* (2.2.2)*) under $\mathbb{P}_0(\cdot \mid \mathcal{D})$. Now apply Lemma 2.4.2, valid for any $\mathcal{D}$ since H4 a) holds for all g^B owing to the choice $\zeta = \zeta_\infty$. As β is bounded, we obtain*

$$|\chi_\infty - \chi_\star| \le \frac{1}{1-\alpha}\left(C_\beta\left(\mathbb{E}_0\left(|\widehat{\Psi^B}(Z_1) - \Psi(Z_1)|^2 \mid \mathcal{D}\right)\right)^{1/2} + \left(\frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/2}\right), \tag{2.4.13}$$

$$|\xi_\infty - \xi_\star| \le 2^{1/3}\left(2C_f + 1\right) \sup_{x\in[\xi_\star-\zeta_\infty,\xi_\star+\zeta_\infty]} |f(F^{-1}(x))|^{-1} \tag{2.4.14}$$
$$\times\left(C_\beta^2\mathbb{E}_0\left(|\widehat{\Psi^B}(Z_1) - \Psi(Z_1)|^2 \mid \mathcal{D}\right) + \frac{\mathbb{V}\mathrm{ar}_0(\psi)}{M'}\right)^{1/3}.$$

Now, write

$$\mathbb{E}_0\left(|\widehat{\Psi^B}(Z_1) - \Psi(Z_1)|^2 \mid \mathcal{D}\right) \le 2\mathbb{E}_0\left(|\widehat{\Psi^B}(Z_1) - \Psi^B(Z_1)|^2 \mid \mathcal{D}\right) + 2\mathbb{E}_0\left(|\Psi^B(Z_1) - \Psi(Z_1)|^2\right).$$

Note that the first expectation of the right hand side is exactly controlled using Theorem 2.7.4. For the second term, write

$$|\Psi^B(Z_1) - \Psi(Z_1)| \le \mathbb{E}_0(|-B \vee \psi \wedge B - \psi| \mid Z_1) \le \mathbb{E}_0((|\psi| - B)^+ \mid Z_1). \tag{2.4.15}$$

We now easily obtain the desired estimates by taking the expectation in (2.4.13)-(2.4.14)*, applying Theorem 2.7.4 and using* (2.4.15)*, together with* $\mathbb{E}(|Z|^{1/p}) \le (\mathbb{E}(|Z|))^{1/p}$ *for any $p \ge 1$.*

2.5 Risk Margin

2.5.1 Dynamization of the Setup

Let there be given an $\mathbb{R}^q$ valued process $Z = \{Z_t, t \ge 0\}$, with $Z_0 = z$, non-homogeneous Markov in its own filtration on our probability space $(\Omega, \mathcal{A}, \mathbb{P})$. The process Z plays the role of observable risk factors. Conditional probabilities, expectations, value-at-risks and expected shortfalls at a level $a \in (0,1)$, given Z_t, are denoted by $\mathbb{P}_t$, $\mathbb{E}_t$, $\mathbb{V}\mathrm{a}\mathbb{R}_t^a$, and $\mathbb{ES}_t^a$. Other sources of randomness arising in $(\Omega, \mathcal{A}, \mathbb{P})$ may be unobservable factors (like hidden financial variables, private information). We assume that Z can be simulated exactly (in other words, we ignore for the sake of simplicity a vanishing time discretization bias regarding Z, which could be considered without major difficulty). We denote by $\bar{Z}_t = (t, Z_t)$ the time-homogenized Markov extension of Z. We write $Z_{[s,t]}$ and $\bar{Z}_{[s,t]}$ for the paths of Z and $\bar{Z}$ on the interval $[s,t]$. We define the discount factor

$$\beta(\bar{Z}_{[0,t]}) := e^{-\int_0^t r(\bar{Z}_s)\mathrm{d}s},$$

for some bounded from below, continuous interest rate function r (hence, in particular, a bounded discount factor). We may then consider the following specification of (2.2.1) :

$$L := \phi(\bar{Z}_{[0,1]}) + \beta(\bar{Z}_{[0,1]})\mathbb{E}_1\left[\psi(\bar{Z}_{[1,T]})\right] - \mathbb{E}_0\left[\psi'(\bar{Z}_{[0,T]})\right], \tag{2.5.1}$$

where ϕ and ψ are real valued measurable functions.

Remark 2.5.1 *The functions ϕ and ψ could depend on variables other than Z, it would not have any significant impact on the analysis.*

For instance, we could consider a Euro Median Term Note (EMTN), issued by a bank, with a performance linked to the 1 year Euribor rate denoted by Z ; then the cashflow for the bank may take the form $\varphi(Z_1)\mathbf{1}_{\tau\geq 1} = \psi'(\bar{Z}_{[0,1]}, \tau)$, where τ is the default time for the bank (assumed independent from Z for simplicity).

In the regression setup of Algorithm 3, this flexibility of using "Z smaller than an underlying high-dimensional factor process" allows embedding in our framework the common industry practice of "partial regressions" with respect to reduced sets of factors.

More broadly, let, for $t \geq 0$ (cf. (2.5.1) for $t = 0$),

$$L^t_{t+1} := \phi(\bar{Z}_{[t,t+1]}) + \beta(\bar{Z}_{[t,t+1]})\,\mathbb{E}_{t+1}[\psi(\bar{Z}_{[t+1,T]})] - \mathbb{E}_t[\psi(\bar{Z}_{[t,T]})]. \tag{2.5.2}$$

Let $\mathbb{V}a\mathbb{R}^a_t[L^t_{t+1}]$ denote a value-at-risk at level $a \in \left(\frac{1}{2}, 1\right)$ of L^t_{t+1} for the conditional distribution[5] of L^t_{t+1} given $\mathcal{F}_t$, i.e.

$$\mathbb{P}_t\left(L^t_{t+1} > \mathbb{V}a\mathbb{R}^a_t[L^t_{t+1}]\right) = 1 - a.$$

Let

$$\mathrm{ES}(\bar{Z}_t) := (1-\alpha)^{-1}\int_\alpha^1 \mathbb{V}a\mathbb{R}^a_t[L^t_{t+1}]\,\mathrm{d}a \tag{2.5.3}$$

denote the corresponding expected shortfall of (fixed) level $\alpha \in \left(\frac{1}{2}, 1\right)$.

2.5.2 Theoretical Risk Margin Estimate

The risk margin RM (called KVA in banking parlance) estimates how much it would cost the firm (bank or insurance) to remunerate its shareholders at the hurdle rate $h > 0$ (e.g. 10%) for their capital at risk $\mathrm{ES}(\bar{Z}_t)$ at any future time t (see Section 2.1). Given the final maturity T of the portfolio, the corresponding formula in [12] reads as

$$\mathrm{RM} = h\mathbb{E}\left[\int_0^T e^{-ht}\beta_t \mathrm{ES}(\bar{Z}_t)dt\right] = \mathbb{E}\left[\beta_\zeta \mathrm{ES}(\bar{Z}_\zeta)\,\mathbf{1}_{\zeta\leq T}\right], \tag{2.5.4}$$

where the second equality follows by randomization of the integral with an independent exponential time ζ of parameter h.

Accordingly, we propose the risk margin estimator

$$\mathbb{E}\left[\beta_\zeta \mathrm{ES}(\bar{Z}_\zeta)\,\mathbf{1}_{\zeta\leq T}\right] \approx \frac{1}{N}\sum_{n=1}^N \beta_{\zeta^n}\widehat{\mathrm{ES}}(\bar{Z}^n_{\zeta^n})\mathbf{1}_{\zeta^n\leq T}, \tag{2.5.5}$$

where $\{\bar{Z}^n_{\zeta^n}, n \geq 1\}$ are independent random variables with the same distribution as $\bar{Z}_\zeta$ and where $\widehat{\mathrm{ES}}(\cdot)$ is one of the estimators of $\mathrm{ES}(\cdot)$ considered in the previous sections, now made conditional on Z_t.

The convergence of the ensuing estimator to the risk margin obtained by sampling an outer expectation of inner conditional expected shortfall estimates could be established by taking an outer expectation of the errors for $\widehat{\mathrm{ES}}(\bar{Z}^n_{\zeta^n})$ estimates of

[5] assumed atomless, cf. Assumptions 4.a), 5, and 4 a) again in the basic, nested, and regressed setups of Algorithms 1, 2, and 3, respectively.

the $\mathrm{ES}(\bar{Z}^n_{\varsigma^n})$ in (2.5.5), errors obtained from the conditional version of the results of Sections 2.3 and 2.4 (or, more precisely, of the awaited but technical developments of these results in terms of convergence rates). By contrast, how to "make conditional" the convergence arguments of [68] or [37] and "aggregate them" to establish the convergence of an outer risk margin estimate is far from clear.

2.5.3 KVA Case Study

Our case study is based on the setup of Armenti and Crépey [24], Section 4 (see also Section 4.4 in [2]), which we recall as a starting point. We consider a clearing house (or central counterparty, CCP for short) with a finite number (≥ 2) of clearing members labeled by i. We denote by :

- T : an upper bound on the maturity of all claims in the CCP portfolio, also accounting for a constant time $\delta > 0$ of liquidating the positions of defaulting clearing members ;
- D^i_t : The cumulative contractual cash flow process of the CCP portfolio of the member i, cash flows being counted positively when they flow from the clearing member to the CCP ;
- $\mathrm{MtM}^i_t = \mathbb{E}_t[\int_t^T \beta_t^{-1}\beta_s dD^i_s]$: The mark-to-market of the CCP portfolio of the member i ;
- τ_i, $\tau_i^\delta = \tau_i + \delta$ and $\boldsymbol{\delta}_{\tau_i^\delta}(dt)$: The default and liquidation times of the member i, a Dirac measure at time τ_i^δ ;
- $\Delta^i_{\tau_i^\delta} = \int_{[\tau_i,\tau_i^\delta]} \beta_t^{-1}\beta_s dD^i_s$: The cumulative contractual cash flows of the member i, accrued at the OIS rate, over the liquidation period of the clearing member i ;
- IM^i_t : The initial margin (IM) posted by the member i as a guarantee in case it defaults, given at time t as a conditional value-at-risk (at a given confidence level a_{ma}) of $\beta_t^{-1}\big(\beta_{t+\delta}(\mathrm{MtM}^i_{t+\delta} + \Delta^i_{t+\delta}) - \mathrm{MtM}^i_t\big)^+$.

Beyond the first ring of defense provided by initial margin (and, of course, variation margin, which we assume equal to the process MtM^i_t stopped at time τ_i), a CCP maintains an additional resource, known as the default fund, against extreme and systemic risk. The current EMIR regulation sizes the default fund of a CCP by the Cover 2 rule, i.e. enough to cover the joint default of the two clearing members with the greatest CCP exposures, which purely relies on market risk. By contrast, we consider in the setup of this case study a broader risk-based specification, in the form of an economic capital of the CCP, which would be defined as a conditional expected shortfall, at some confidence level a_{df}, of its one-year ahead loss-and-profit if there was no default fund, as it results from the combination of the credit risk of the clearing members and of the market risk of their portfolios. As developed in [24], such a specification can be used for allocating the default fund between the clearing members, after calibration of the quantile level a_{df} to the Cover 2 regulatory rule at time 0.

Specifically, we define the loss process of a CCP that would be in charge of dealing with member counterparty default losses through a CVA^{ccp} account (earning the risk-free rate r) as, for $t \in (0, T]$ (starting from some arbitrary initial value, since it is only the fluctuations of L^{ccp} that matter in what follows),

$$\begin{aligned}\beta_t dL_t^{ccp} = \sum_i \big(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{MtM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\big)^+ \boldsymbol{\delta}_{\tau_i^\delta}(dt)\\ + \beta_t(d\mathrm{CVA}_t^{ccp} - r_t \mathrm{CVA}_t^{ccp})dt,\end{aligned} \tag{2.5.6}$$

where the CVA of the CCP is given as

$$\mathrm{CVA}_t^{ccp} = \mathbb{E}_t \sum_{t<\tau_i^\delta<T} \beta_t^{-1}\big(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{MtM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\big)^+,\ 0 \le t \le T \tag{2.5.7}$$

(in particular, L^{ccp} is constant from time T onward).

We define the corresponding economic capital process of the CCP as

$$\mathrm{EC}_t^{ccp} = \mathbb{ES}_t^{a_{df}}\left(\int_t^{t+1} \beta_t^{-1}\beta_s dL_s^{ccp}\right), \tag{2.5.8}$$

where, by (3.4.10),

$$\begin{aligned}\beta_t^{-1}\int_t^{t+1} \beta_s dL_s^{ccp} = \beta_t^{-1} \sum_{t<\tau_i^\delta\le t+1} \big(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{MtM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\big)^+\\ + (\beta_t^{-1}\beta_{t+1}\mathrm{CVA}_{t+1}^{ccp} - \mathrm{CVA}_t^{ccp}).\end{aligned} \tag{2.5.9}$$

The KVA (or risk margin) of the CCP estimates how much it would cost the CCP to remunerate all clearing members at some hurdle rate h for their capital at risk in the default fund from time 0 onward, namely, assuming a constant interest rate r (cf. (2.5.4), (3.4.11), and [24]) :

$$\mathrm{KVA}^{ccp} = h\mathbb{E}\left[\int_0^T e^{-(r+h)s}\mathrm{EC}_s^{ccp} ds\right]. \tag{2.5.10}$$

For our numerics we consider the CCP toy model of Section 4 in [24] and Section 4.4 in [2], where nine members are clearing (interest rate or foreign exchange) swaps on a Black–Scholes underlying rate process X, with all the numerical parameters used there. The default times of the clearing members are defined through the common shock model or dynamic Marshall-Olkin copula (DMO) model of [34], Chapt. 8–10 and [49] (see also [52, 55]).

Mapping with the General Setup

This model, where defaults can happen simultaneously with positive probabilities, results in a Markovian pair $Z = (X, Y)$ made of, on the one hand, the underlying Black–Scholes rate X and, on the other hand, the vector Y of the default indicator processes of the clearing members. As a consequence, all conditional expectations, value-at-risks (embedded in the IM^i numbers), and expected shortfalls (embedded in the EC^{ccp} numbers) are functions of the pair (t, Z), so that, with $Z = (X, Y)$ as above, we can identify

$$\begin{aligned} L_{t+1}^t &\leftarrow \int_t^{t+1} \beta_t^{-1}\beta_s dL_s^{ccp},\\ \mathrm{ES}(t,z) &\leftarrow \mathbb{ES}^{a_{df}}\left(\int_t^{t+1} \beta_t^{-1}\beta_s dL_s^{ccp} \,\middle|\, Z_t = z\right).\end{aligned}$$

The ensuing KVA can be computed by Algorithms 1 (for validation purposes, building on the explicit CVA^{ccp} formulas that are available in our stylized setup, cf. [24, Section A]), 2, or 3 for the inner EC^{ccp} computations, which are then aggregated as explained above. However, for GPU optimization reasons developed in [2, Appendices A and B], we do not rely on the randomized version (given by the right-hand side formulation) of the risk margin in (2.5.4), i.e. we do not use the unbiased estimator (2.5.5), resorting instead on a Riemann sum approximation of step six months of the time integral that is visible in the left-hand side in (2.5.4).

Depending on the algorithm that is used, we can identify further $\beta(\bar{Z}_{[s,u]}) = e^{-r(u-s)}$ and :

- In the case of Algorithm 1 :

$$\phi(\bar{Z}[t,t+1]) \leftarrow L^t_{t+1}, \qquad \psi \leftarrow 0;$$

- In the case of Algorithms 2 or 3 :

$$\phi(\bar{Z}[s,u]) = \psi(\bar{Z}_{[s,u]}) \leftarrow \beta_s^{-1} \sum_{s<\tau_i^\delta \le u} \big(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{MtM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\big)^+. \tag{2.5.11}$$

With respect to the general setup of previous sections, the methodological assumptions, such as the ones on the sequences γ_k of the SA parameters or the requirement made in H11 of using a regression sample independent from the rest of the simulation in the context of Algorithm 3, can always be met at implementation stage.

Regarding now the abstract assumptions there, we only make a general comment that they should all hold in our lognormal model for X combined with randomized sampling at the times of defaults of the counterparties, which are all times with an intensity, recalling the corresponding modeling assumptions related to Algorithms 1 (SA scheme for the basic case without liabilities), 2 (SA scheme with nested simulation of future liabilities) and Algorithm 3 (SA scheme with regression of future liabilities), respectively :

- H4 a) [continuous cdf of the loss L], H4 b) [second moment of L],
- H5 [density and second moment of the loss] (the density part should follow from Malliavin calculus considerations), H6 [moments of order ≥ 2 of the (present and future) liabilities in L], H8 [concentration inequalities related to (and implying exponential moments of) the present and future liabilities in L], H9 [second moment of L, moments of order > 2 of the (present and future) liabilities in L] ;
- H4 a) again, H11 [square integrability of future loss components], H12 [continuous cdf of the loss where one replaces the future liability function Ψ by an arbitrary regression basis function $g \in \widehat{\mathcal{H}}$], H2.4.1 [bounded density of L, positive and continuous on an interval (specified further in H13) around $\xi_\star$].

Regarding the regression algorithm for CVA$^{ccp}_{t+1}$ that is required in the context of Algorithm 3, we apply to CVA$^{ccp}_{t+1} = \mathbb{E}_{t+1}\psi(\bar{Z}_{[t+1,T]})$ the approach that is used for computing the "CA process" in Section 4.4 of [2], using as a regression basis 1, X_{t+1}, X^2_{t+1} (recall X is the underlying Black–Scholes rate) and the default indicator processes at time $(t+1)$ of the clearing members. In the present case of CVA$^{ccp}_{t+1}$ the situation is in fact a bit simpler as no time-stepping is required, i.e. we just need one regression for each time $(t+1)$ that occurs via the discretization times t of the integral visible in (2.5.4), because CVA^{ccp}_{t+1} is a conditional expectation, as opposed to the above-mentioned CA process, which only solves a semi-linear BSDE.

For an SA scheme launched at time t of the outer KVA simulation, we use $\gamma_k = \frac{\gamma_0}{(100+k^{0.75})} \times \frac{(T-t)}{T}$, starting from the initial condition $\xi_0 = \chi_0 = 0$.

Numerical Results

All our simulations are run on a laptop that has an Intel i7-7700HQ CPU and a single GeForce GTX 1060 GPU programmed with the CUDA/C application programming interface (API).

Table 2.1 shows the time 0 (unconditional) expected shortfalls over the first year, obtained by four variants of the SA scheme and for three levels of the quantile a_{df}.

K	a_{df}	$\mathrm{ES}^{(a)}$	$\mathrm{ES}^{(b)}$	$\mathrm{ES}^{(c)}$	$\mathrm{ES}^{(d)}$
10^4	85%	311.23	248.05	253.11	259.13
	95.5%	924.72	924.72	924.72	924.72
	99%	2406.77	2406.77	2406.77	2406.77
10^5	85%	296.24	202.27	207.81	211.32
	95.5%	858.72	858.72	858.72	858.72
	99%	2347.83	2347.83	2347.83	2347.83
5×10^5	85%	287.85	200.12	206.05	209.37
	95.5%	849.12	849.12	849.12	849.12
	99%	2327.45	2327.45	2327.45	2327.45

TABLE 2.1 – Time 0 unconditional expected shortfalls, computed : [$\mathrm{ES}^{(a)}$] by Algorithm 1 without the CVA terms, i.e. forgetting about the second line in (3.4.12), in order to assess, by comparison with the other results, the impact of these CVA terms on economical capital, depending on the confidence level a_{df} ; [$\mathrm{ES}^{(b)}$] by Algorithm 1 with the CVA terms computed by the explicit formulas that are available in the lognormal market model of this case study ; [$\mathrm{ES}^{(c)}$] by Algorithm 2 with the CVA_1 terms computed by nested Monte Carlo (and CVA_0 computed by outer Monte Carlo) ; [$\mathrm{ES}^{(d)}$] by Algorithm 3 with the CVA_1 terms computed by regression against X_1 and the default indicators of the clearing members at time 1 (and CVA_0 computed by outer Monte Carlo).

In the case $a_{df} = 85\%$, Figure 2.1 shows the corresponding (time discretized) ES processes obtained after $K = 10^4$ and $K = 5 \times 10^5$ iterations of the SA schemes ;

K	$\mathrm{KVA}^{(a)}$	$\mathrm{KVA}^{(b)}$	$\mathrm{KVA}^{(c)}$	$\mathrm{KVA}^{(d)}$
10^4	66.70	32.62	34.71	37.31
10^5	57.54	24.09	26. 85	29.11
5×10^5	54.89	23.57	25.08	28.76

TABLE 2.2 – KVA for $a_{df} = 85\%$ and $N = 1024$ outer trajectories.

2.6 Conclusion and Perspectives

In this paper we propose convergent stochastic approximation estimators for the economic capital of a loss random variable L that entails a future liability (conditio-

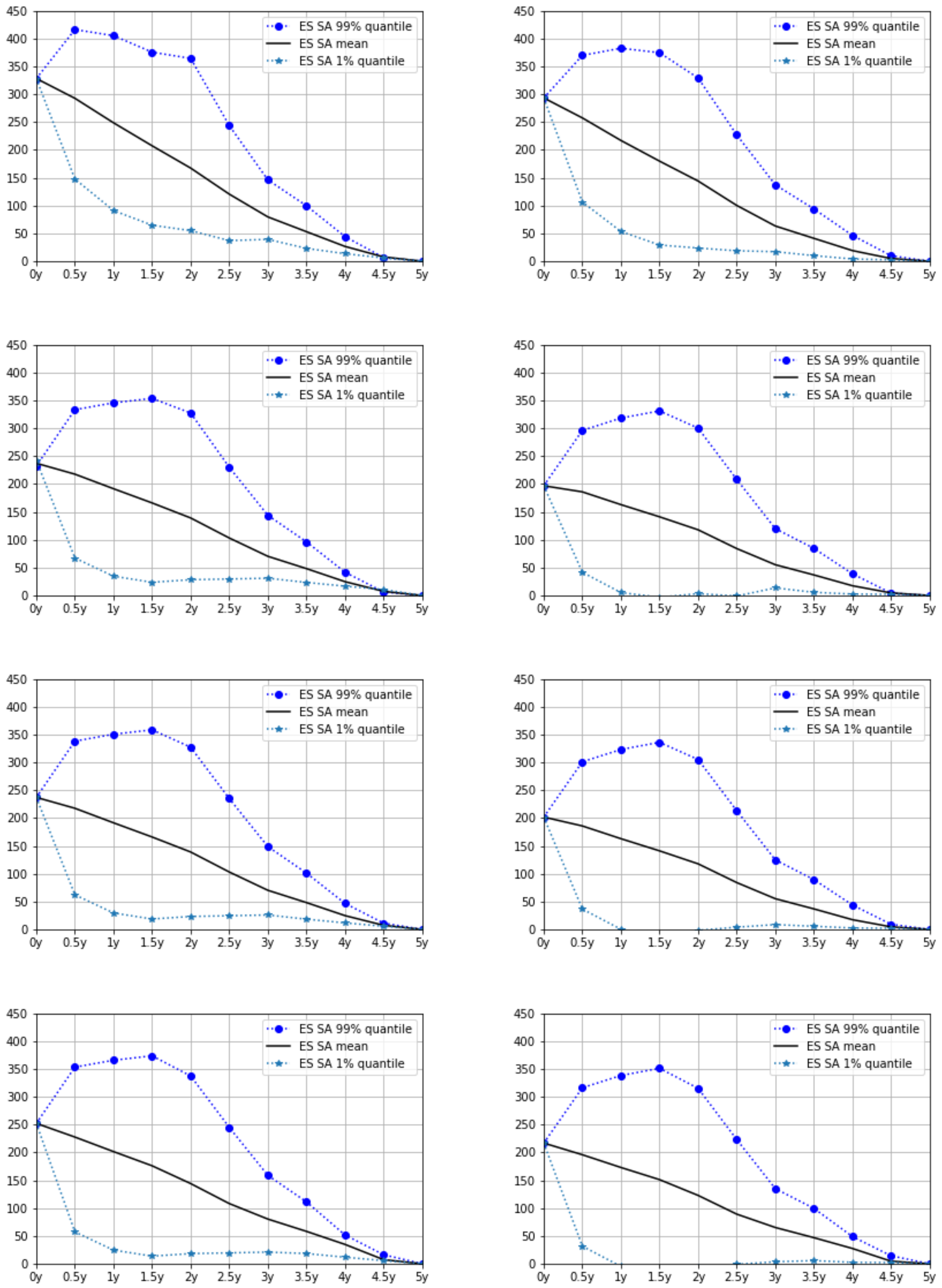


FIGURE 2.1 – $\mathrm{ES}^{(a)}, \mathrm{ES}^{(b)}, \mathrm{ES}^{(c)}$, and $\mathrm{ES}^{(d)}$ processes (top to bottom) for $a_{df} = 85\%$ and $N = 1024$ outer trajectories. Left : $K = 10^4$. Right : $K = 5 \times 10^5$.

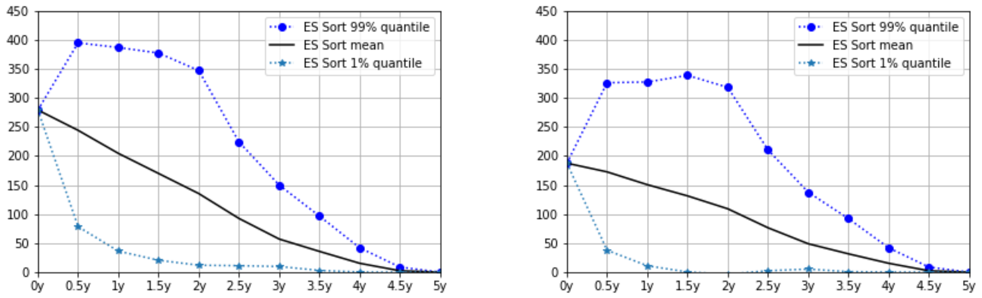


Figure 2.2 – ES benchmark process obtain by tri method for $a_{df} = 85\%$, $N = 1024$ outer trajectories and $M_{ec} = 32 * 100$ inner trajectorires. Left : case without CVA terms. Right : case with CVA terms computed by the explicit formulas.

nal expectation). The latter is estimated either by nested Monte Carlo as in Gordy and Juneja [68], or by regression as in Broadie, Du, and Moallemi [37]. Then we embed conditional versions of the above into outer risk margin (or KVA) computations.

From a practical point of view, an incremental SA scheme uses a limited amount of memory but, being a loop, is less easy to parallelize than a simulation-and-sort algorithm, on which several processors can fruitfully be used (see [2, Appendix C]). On the other hand SA schemes can be efficiently combined with importance sampling as studied in [28,29], whereas [68] and [37] introduce respective jacknife and weighted regression acceleration procedures for the simulation-and-sort schemes.

From a theoretical point of view, the stochastic approximation viewpoint leads to stronger convergence results under considerably smoother assumptions than together [68] and [37]. In particular, our assumptions (recalled in Section 2.5.3) only bear on the limiting problem, as opposed to unverifiable (not to say implausible) assumptions on the perturbed approximating problems in [68] and [37] :

- Assumptions on the density of the nested Monte Carlo surrogate of the loss in [68] ;
- Invertibility of the empirical covariance matrix of the regressors and an orthonormal basis of empirical regressors in [37]. By contrast, we do not even need to assume a vector space of theoretical regressors ; for instance, our space of theoretical regressors could be given in the form of a neural network.

About now the results :

- [68] only shows mean square convergence, whereas we show almost sure convergence ;
- [37] considers a very stylized proxy of expected shorfall in the form of $\mathbb{E}(L - \xi)^+$, for a known and fixed ξ, instead of the value-at-risk of L that needs to be estimated in the first place in a genuine expected shorfall perspective. Moreover, their study is asymptotic in the number of simulations for a fixed number of basis functions, they do not address the global convergence problem when the size of the regression basis and the number of simulations jointly go to infinity.

Last, regarding the comparison between the stochastic approximation schemes with nested versus regressed estimation of future liabilities, the assumptions that allow establishing the convergence of either approach are discussed and compared along the paper. In order to compare the fine convergence properties of each approach, it would be useful to push the computations to obtain the L_2 errors in both cases, which we leave for further research.

2.7 Technical Developments

We denote by $|x|$ the (Euclidean) norm of $x \in \mathbb{R}^d$ and by $\langle x, y\rangle$ the inner product of two vectors $x, y \in \mathbb{R}^d$. Vectors $x \in \mathbb{R}^d$ are column-vectors, and A^T denotes the transpose of a matrix A.

Some of the results are general (not specific to the setup of the main body of the paper) and therefore stated in terms of an abstract probability measure $\mathbb{Q}$, with related expectation denoted by $\mathbb{E}$.

2.7.1 Two Identities

Lemma 2.7.1 *Let $\alpha \in (0,1)$ and μ be a probability distribution on $\mathbb{R}$ having a first order moment. If $\xi_\infty \neq \xi'_\infty$ are two solutions of $\xi : 1-\alpha = \int_\xi^\infty \mu(\mathrm{d}x)$ then*

$$\xi_\infty + \frac{1}{1-\alpha}\int (x-\xi_\infty)^+ \mu(\mathrm{d}x) = \xi'_\infty + \frac{1}{1-\alpha}\int (x-\xi'_\infty)^+ \mu(\mathrm{d}x).$$

Proof *We can assume that $\xi_\infty < \xi'_\infty$. Since*

$$\int_{\xi_\infty}^{+\infty} \mu(\mathrm{d}x) = \int_{\xi'_\infty}^{+\infty} \mu(\mathrm{d}x) = 1-\alpha,$$

then $\int_{\xi_\infty}^{\xi'_\infty} \mu(\mathrm{d}x) = 0$. Upon noting that

$$\xi_\infty + \frac{1}{1-\alpha}\int (x-\xi_\infty)^+ \mu(\mathrm{d}x) = \frac{1}{1-\alpha}\int_{\xi_\infty}^{+\infty} x\,\mu(\mathrm{d}x)$$

we obtain

$$\xi_\infty + \frac{1}{1-\alpha}\int (x-\xi_\infty)^+ \mu(\mathrm{d}x) = \xi'_\infty + \frac{1}{1-\alpha}\int (x-\xi'_\infty)^+ \mu(\mathrm{d}x).$$

Lemma 2.7.2 *Let $\{\gamma_k, k \geq 1\}$ be a positive sequence such that $\sum_k \gamma_k = +\infty$ and $\lim_k \gamma_k = 0$. Given a $\mathbb{R}^d$-valued sequence $\{V_k, k \geq 1\}$ and $\theta_0 \in \mathbb{R}^d$, define the sequence $\{\theta_k, k \geq 0\}$ by*

$$\theta_{k+1} = (1-\gamma_{k+1})\theta_k + \gamma_{k+1}V_{k+1}.$$

Set $S_0 = 1$ and $S_k := \prod_{j=1}^k (1-\gamma_j)^{-1}$ for any $k \geq 1$. Then $\lim_k S_k = +\infty$ and when it exists,

$$\lim_k \theta_k = \lim_k \frac{1}{S_k}\sum_{j=1}^k S_j \gamma_j V_j.$$

Proof *The proof is adapted from [28]. Note that $S_{k+1}(1-\gamma_{k+1}) = S_k$, hence, for all k large enough such that $\gamma_k \to 0$, $\ln S_{k+1} + \ln(1-\gamma_{k+1}) = \ln S_k$, i.e.*

$$\ln S_{k+1} - \ln S_k = -\ln(1-\gamma_{k+1}) \geq \gamma_{k+1}.$$

Since $\sum_k \gamma_k = \infty$*, then* $\lim_k S_k = +\infty$*.*

We write

$$S_{k+1}\theta_{k+1} = S_{k+1}(1-\gamma_{k+1})\theta_k + S_{k+1}\gamma_{k+1}V_{k+1} = S_k\theta_k + S_{k+1}\gamma_{k+1}V_{k+1}.$$

Hence, by induction, $S_{k+1}\theta_{k+1} = S_0\theta_0 + \sum_{j=1}^{k+1} S_j\gamma_j V_j$*, which implies the result and the conclusion follows from this.*

2.7.2 A General Convergence Result for Stochastic Approximation Algorithms

Let $H : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ be a measurable function and let $\{\gamma_k, k \geq 1\}$ be a sequence of positive numbers. Let $\mathbb{R}^q$-valued random variables $\{V_k, k \geq 0\}$ and $\theta_0 \in \mathbb{R}^d$ be defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Theorem 2.7.1 provides sufficient conditions for the almost-sure convergence and the L^p-convergence, $p \in (0,2)$, of the sequence $\{\theta_k, k \geq 0\}$ given by

$$\theta_{k+1} = \theta_k - \gamma_{k+1}\ H(\theta_k, V_{k+1}). \tag{2.7.1}$$

These conditions are general enough to cover the case when the r.v. $\{V_k, k \geq 1\}$ are not i.i.d. but have a distribution converging, in some sense, to the distribution of a r.v. $V_\star$.

We write

$$H(\theta_k, V_{k+1}) = h(\theta_k) + e_{k+1} + r_{k+1}, \tag{2.7.2}$$

where

$$\begin{aligned} h(\theta) &:= \mathbb{E}\left[H(\theta, V_\star)\right], \\ e_{k+1} &:= H(\theta_k, V_{k+1}) - \mathbb{E}\left[H(\theta_k, V_{k+1})|\mathcal{G}_k\right], \\ r_{k+1} &:= \mathbb{E}\left[H(\theta_k, V_{k+1})|\mathcal{G}_k\right] - h(\theta_k), \end{aligned}$$

and where the filtration $\{\mathcal{G}_k, k \geq 1\}$ is defined by $\mathcal{G}_k := \sigma\{V_1, \cdots, V_k\}$.

Theorem 2.7.1 *Suppose that*

(i) $\{\gamma_k, k \geq 1\}$ *is a deterministic positive sequence such that* $\sum_k \gamma_k = +\infty$ *and there exists* $\kappa \in (0,1]$ *such that* $\sum_{k\geq 1} \gamma_k^{1+\kappa} < \infty$*,*

(ii) $H : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ *is measurable and* $h : \mathbb{R}^d \to \mathbb{R}^d$ *is continuous,*

(iii) the set $\mathcal{L} := \{h = 0\}$ *is a non-empty compact subset of* $\mathbb{R}^d$ *and for any* $\theta^* \in \mathcal{L}$ *and* $\theta \notin \mathcal{L}$*, we have* $\langle \theta - \theta^*, h(\theta)\rangle > 0$*.*

Let $\{\theta_k, k \geq 0\}$ *be given by* (2.7.1) *where the r.v.* $\{V_k, k \geq 0\}$ *satifsy*

(iv) $\sum_{k\geq 1} \gamma_k^{1-\kappa}|r_k|^2 < +\infty$ $\mathbb{P}$*-a.s.*

(v) There exist non-negative constants $C_{H,1}, C_{H,2}$ *such that, for any* $\theta \in \mathbb{R}^d$*,*

$$\sup_{k\geq 1} \mathbb{E}\left[|H|^2(\theta, V_k)\right] \leq C_{H,1} + C_{H,2}|\theta|^2.$$

Then there exists a $\mathcal{L}$*-valued random variable* θ_∞ *such that* $\mathbb{P}(\lim_k \theta_k = \theta_\infty) = 1$*. If, in addition,*

(vi) $\sum_{k\geq 1} \gamma_k^{1-\kappa}\mathbb{E}[|r_k|^2] < +\infty$ *,*

then $\sup_{k\geq 0} \mathbb{E}[|\theta_k - \theta_\infty|^2] < +\infty$ *and for any* $p \in (0,2)$*,* $\lim_k \mathbb{E}\left[|\theta_k - \theta_\infty|^p\right] = 0$*.*

Proof **Step 1. Almost-sure boundedness and convergence.** *Let $\theta^* \in \mathcal{L}$. We have, by* (2.7.2),

$$\begin{aligned}|\theta_{k+1}-\theta^*|^2 &= |\theta_k-\theta^*-\gamma_{k+1}\left(h(\theta_k)+e_{k+1}+r_{k+1}\right)|^2\\ &= |\theta_k-\theta^*|^2-2\gamma_{k+1}\langle\theta_k-\theta^*,h(\theta_k)\rangle\\ &\quad -2\gamma_{k+1}\langle\theta_k-\theta^*,e_{k+1}\rangle-2\gamma_{k+1}\langle\theta_k-\theta^*,r_{k+1}\rangle+\gamma_{k+1}^2|H|^2(\theta_k,V_{k+1}).\end{aligned}$$

Since $\{e_k, k\geq 1\}$ is a martingale-increment w.r.t. the filtration $\{\mathcal{G}_k, k\geq 1\}$ and θ_k is $\mathcal{G}_k$-measurable, we have for any k,

$$\begin{aligned}\mathbb{E}\left[|\theta_{k+1}-\theta^*|^2|\mathcal{G}_k\right] \leq |\theta_k-\theta^*|^2-2\gamma_{k+1}\langle\theta_k-\theta^*,h(\theta_k)\rangle+\gamma_{k+1}^{1+\kappa}|\theta_k-\theta^*|^2+\gamma_{k+1}^{1-\kappa}|r_{k+1}|^2\\ +\gamma_{k+1}^2C_{H,1}+\gamma_{k+1}^2C_{H,2}|\theta_k|^2,\end{aligned}$$

where we used the inequality $-2\gamma\langle a,b\rangle\leq\gamma^{1+\kappa}|a|^2+\gamma^{1-\kappa}|b|^2$, the equality $\mathbb{E}[r_{k+1}|\mathcal{G}_k]=r_{k+1}$ and the assumption (v)*. Hence, by using $|\theta_k|^2\leq 2|\theta_k-\theta^*|^2+2|\theta^*|^2$,*

$$\mathbb{E}\left[|\theta_{k+1}-\theta^*|^2|\mathcal{G}_k\right]\leq\left(1+2\gamma_{k+1}^2C_{H,2}+\gamma_{k+1}^{1+\kappa}\right)|\theta_k-\theta^*|^2 \qquad (2.7.3)$$
$$-2\gamma_{k+1}\langle\theta_k-\theta^*,h(\theta_k)\rangle+\gamma_{k+1}^{1-\kappa}|r_{k+1}|^2+\gamma_{k+1}^2C', \qquad (2.7.4)$$

where $C':=C_{H,1}+2C_{H,2}|\theta^*|^2$. From the assumptions (i), (iii)*, and* (iv)*, we have that, $\mathbb{P}$-a.s.,*

$$\forall k\geq 0\quad \gamma_{k+1}\langle\theta_k-\theta^*,h(\theta_k)\rangle\geq 0,\qquad \sum_{k\geq 0}(\gamma_{k+1}^{1-\kappa}|r_{k+1}|^2+\gamma_{k+1}^2+\gamma_{k+1}^{1+\kappa})<+\infty.$$

By the Robbins-Siegmund lemma (see [108]), $\mathbb{P}$-a.s. (for an almost-sure set depending upon θ^*)

$$\lim_k|\theta_k-\theta^*| \text{ exists},\qquad \sum_{k\geq 0}\gamma_{k+1}\langle\theta_k-\theta^*,h(\theta_k)\rangle<+\infty.$$

Since $\mathcal{L}$ is bounded and $\theta^*\in\mathcal{L}$, this implies that the sequence $\{\theta_k, k\geq 0\}$ is bounded with probability one. Using the separability of $\mathbb{R}^d$ and since $\theta'\mapsto\lim_k|\theta_k-\theta'|$ is continuous, we have $\mathbb{P}$-a.s. :

$$\forall\,\theta'\in\mathcal{L},\quad \text{the limit } \lim_k|\theta_k-\theta'| \text{ exists.} \qquad (2.7.5)$$

Set $\varsigma:=\liminf_{k\to+\infty}\langle\theta_k-\theta^*,h(\theta_k)\rangle$. By (iii)*, $\varsigma\geq 0$. As $\sum_{k\geq 0}\gamma_k=+\infty$, we have*

$$\{\varsigma>0\}\subseteq\sum_{k\geq 0}\gamma_{k+1}\langle\theta_k-\theta^*,h(\theta_k)\rangle=+\infty;$$

the probability of the second event is zero. Hence $\mathbb{P}(\varsigma=0)=1$.

Therefore, with probability one, there exists a subsequence $\{n_k, k\geq 1\}$ such that $\lim_k\langle\theta_{n_k}-\theta^*,h(\theta_{n_k})\rangle=0$. Since the sequence $\{\theta_{n_k}, k\geq 1\}$ is bounded a.s., we can still assume (up to extraction of another subsequence) that $\{\theta_{n_k}, k\geq 1\}$ converges to some limit θ_∞. By assumption (ii)*, we have $\langle\theta_\infty-\theta^*,h(\theta_\infty)\rangle=0$, and by assumption* (iii)*, this implies that $\theta_\infty\in\mathcal{L}$. But using* (2.7.5) *we get $\lim_k|\theta_k-\theta_\infty|=\lim_k|\theta_{n_k}-\theta_\infty|=0$. This implies that $\lim_k\theta_k=\theta_\infty$.*

Step2. Uniform boundedness in L^2**.** *Let a (deterministic) point $\theta^*\in\mathcal{L}$ be given. By taking expectation in* (2.7.3)*, we have*

$$\mathbb{E}\left[|\theta_{k+1}-\theta^*|^2\right]\leq\left(1+2\gamma_{k+1}^2C_{H,2}+\gamma_{k+1}^{1+\kappa}\right)\mathbb{E}\left[|\theta_k-\theta^*|^2\right]+\gamma_{k+1}^{1-\kappa}\mathbb{E}[|r_{k+1}|^2]+C'\gamma_{k+1}^2.$$

Applying again the Robbins-Siegmund lemma with the assumptions (i) *and* (vi)*, we deduce that the sequence* $\lim_k \mathbb{E}\left[|\theta_k - \theta^*|^2\right]$ *exists and thus* $\sup_k \mathbb{E}\left[|\theta_k|^2\right] < \infty$ *since* $\mathcal{L}$ *is bounded. This implies* $\sup_k \mathbb{E}\left[|\theta_k - \theta_\infty|^2\right] < +\infty$ *for any* $\mathcal{L}$*-valued random variable* θ_∞*, using again that* $\mathcal{L}$ *is bounded.*

Step 3. Convergence in L^p**.** *Let* $C > 0$ *and* $p \in (0,2)$*. We write*

$$\mathbb{E}\left[|\theta_k - \theta_\infty|^p\right] = \mathbb{E}\left[|\theta_k - \theta_\infty|^p \ \mathbf{1}_{|\theta_k - \theta_\infty| < C}\right] + \mathbb{E}\left[|\theta_k - \theta_\infty|^p \ \mathbf{1}_{|\theta_k - \theta_\infty| \geq C}\right].$$

The first term converges to zero by the dominated convergence theorem. For the second term, Hölder's and Markov's inequalities give that

$$\mathbb{E}[|\theta_k - \theta_\infty|^p \mathbf{1}_{|\theta_k - \theta_\infty| \geq C}] \leq \frac{\mathbb{E}[|\theta_k - \theta_\infty|^2]}{C^{2-p}} \leq \frac{\sup_{l \geq 0} \mathbb{E}[|\theta_l - \theta_\infty|^2]}{C^{2-p}};$$

which is lower than $\epsilon > 0$ *for some* C *large enough. This holds true for any* ϵ*, thus concluding the proof.*

2.7.3 Proof of Theorem 2.2.1

For the study of the sequence $\{\xi_k, k \geq 1\}$, we check the assumptions of Theorem 2.7.1, applied to $\theta_k \leftarrow \xi_k$ (so that $d = 1$), $\mathbb{Q} \leftarrow \mathbb{Q}_0$, $V_k \leftarrow L_k$, the distribution of $V_\star$ is $\mathrm{Q}(z, \cdot)$ and

$$H(\theta, V) \leftarrow H_1(\xi, L) = 1 - \frac{1}{1-\alpha} \mathbf{1}_{L > \xi}$$

(cf. (2.2.3)). By H3, the condition (i) is satisfied. In addition, by H4a, the function

$$\xi \mapsto h(\xi) := 1 - \frac{1}{1-\alpha} \int_\xi^{+\infty} \mathrm{Q}(z, \mathrm{d}x)$$

is continuous on $\mathbb{R}$, so that (ii) holds. The set $\mathcal{L} := \{h = 0\}$ is the set of the points $\xi_\star$ satisfying $\mathbb{Q}_0(L > \xi_\star) = 1 - \alpha$: under H4a, this set is non empty and compact and for any $\xi < \xi_\star$ (resp. $\xi > \xi_\star$) such that $h(\xi) \neq 0$, $h(\xi) < 0$ (resp. $h(\xi) > 0$). Hence, (iii) is satisfied. In this algorithm, we have $r_k = 0$ since $h(\cdot) = \mathbb{E}_0\left[H(\cdot, L)\right]$ and L_{k+1} is independent of ξ_k. Hence, (iv) and (vi) hold. Finally, $\sup_{\xi \in R, L \in \mathbb{R}} |H(\xi, L)|^2 \leq C_{H,1} := (1-\alpha)^{-2}$ so that (v) holds with $C_{H,2} = 0$.

For the results on the sequence $\{\chi_k, k \geq 1\}$, we check the assumptions of Lemma 2.7.2 with $\theta_k \leftarrow \chi_k$ (so that $d = 1$) and $V_{k+1} \leftarrow \xi_k + (1-\alpha)^{-1}(L_{k+1} - \xi_k)^+$. We write

$$V_{k+1} = \xi_k + (1-\alpha)^{-1} \int (x - \xi_k)^+ \mathrm{Q}(z, \mathrm{d}x) + \widetilde{e}_{k+1}, \tag{2.7.6}$$
$$\widetilde{e}_{k+1} := (1-\alpha)^{-1} \left((L_{k+1} - \xi_k)^+ - \int (x - \xi_k)^+ \mathrm{Q}(z, \mathrm{d}x) \right).$$

Set $S_0 := 1$ and $S_k := \prod_{j=1}^k (1 - \gamma_j)^{-1}$ for any $k \geq 1$, so that $S_k(1 - \gamma_k) = S_{k-1}$ and $S_k - S_{k-1} = \gamma_k S_k$. By H3 and Lemma 2.7.2, $\lim_k S_k = +\infty$ so that from the above almost-sure convergence on $\{\xi_k, k \geq 1\}$ and from the Cesaro lemma,

$$\lim_{k \to \infty} \frac{1}{S_k} \sum_{j=1}^k S_j \gamma_j \xi_j = \lim_{k \to \infty} \frac{1}{S_k} \sum_{j=1}^k (S_j - S_{j-1}) \xi_j = \xi_\infty \qquad \mathbb{Q}_0 - \text{a.s.}$$

By H4a, the second term in the RHS of (2.7.6) is a continuous function of ξ_k. Therefore, by similar arguments,

$$\lim_{k\to\infty} \frac{1}{S_k} \sum_{j=1}^{k} S_j\gamma_j \int (x-\xi_j)^+ \mathrm{Q}(z, \mathrm{d}x) = \int (x-\xi_\infty)^+ \mathrm{Q}(z, \mathrm{d}x) \qquad \mathbb{Q}_0 - \text{a.s.}$$

Finally, $\{\widetilde{e}_k, k \geq 1\}$ is a $\mathcal{G}_k$-martingale increment ; by using $|(a-c)^+ - (b-c)^+| \leq |a| + |b|$ and $(|a|+|b|)^2 \leq 2a^2 + 2b^2$, and since $\{L_k, k \geq 1\}$ are i.i.d. with distribution $\mathrm{Q}(z, \cdot)$, we have

$$\mathbb{E}_0 \left[\left| (L_{j+1} - \xi_j)^+ - \int (x-\xi_j)^+ \mathrm{Q}(z, \mathrm{d}x) \right|^2 \right] \leq 2 \int x^2 \mathrm{Q}(z, \mathrm{d}x) + 2 \left(\int |x| \mathrm{Q}(z, \mathrm{d}x) \right)^2 ;$$

by H4b, the RHS is finite. Therefore, H3, [74, Corollary 2.2.] and the Kronecker Lemma (see e.g. [74, Section 2.6, page 31]) imply that

$$\lim_{k\to+\infty} \frac{1}{S_k} \sum_{j=1}^{k} S_j\gamma_j \left((L_{j+1} - \xi_j)^+ - \int (x-\xi_j)^+ \mathrm{Q}(z, \mathrm{d}x) \right) = 0, \qquad \mathbb{Q}_0 - \text{a.s.}$$

By Lemma 2.7.2, we obtain that $\mathbb{Q}_0$-a.s., $\lim_k \chi_k$ exists and solves

$$\lim_k \chi_k = \xi_\infty + \frac{1}{1-\alpha} \int (x-\xi_\infty)^+ \mathrm{Q}(z, \mathrm{d}x).$$

2.7.4 Proofs of the Results of Section 2.3.1

We start the proof with two preliminary lemmas.

Lemma 2.7.3 *Let $\{\varphi_m, m \geq 1\}$ be $\mathbb{R}$-valued random variable defined on $(\Omega, \mathcal{A}, \mathbb{P})$, i.i.d. with distribution μ. Assume that μ has a finite moment of order $p_\star > 1$ and set $C_{p_\star} := \int |w - \int w\mu(\mathrm{d}w)|^{p_\star} \mu(\mathrm{d}w)$. Then for any $M \geq 1$,*

$$\mathbb{E}\left[\left| \frac{1}{M} \sum_{m=1}^{M} \varphi_m - \int w\, \mu(\mathrm{d}w) \right|^{p_\star} \right] \leq \frac{c_{p_\star}\, C_{p_\star}}{M^{(p_\star/2)\wedge(p_\star-1)}}, \tag{2.7.7}$$

where $c_p := (18p\sqrt{q})^p$ and q is the Hölder conjugate of p.

Proof *Set $\mu[1] := \int w\mu(\mathrm{d}w)$. The Burkholder inequality (see e.g. [74, Theorem 2.10]) applied to the sequence $\frac{1}{M}(\varphi^m - \mu[1]), m = 1, \ldots, M$, yields*

$$\mathbb{E}\left[\left| \frac{1}{M} \sum_{m=1}^{M} \varphi_m - \mu[1] \right|^{p_\star} \right] \leq \frac{c_{p_\star}}{M^{p_\star}} \mathbb{E}\left[\left| \sum_{m=1}^{M} (\varphi_m - \mu[1])^2 \right|^{p_\star/2} \right].$$

If $p_\star \geq 2$, we obtain by the Minkowsky inequality,

$$\mathbb{E}\left[\left| \sum_{m=1}^{M} (\varphi_m - \mu[1])^2 \right|^{p_\star/2} \right] \leq \left(\sum_{m=1}^{M} \left(\mathbb{E}\left[|\varphi_m - \mu[1]|^{p_\star} \right] \right)^{2/p_\star} \right)^{p_\star/2} \leq C_{p_\star}\, M^{p_\star/2}.$$

If $p_\star < 2$, by using $(x+y)^{p_\star/2} \leq x^{p_\star/2} + y^{p_\star/2}$ for any $x, y \geq 0$, we have

$$\mathbb{E}\left[\left|\sum_{m=1}^{M}(\varphi_m - \mu[1])^2\right|^{p_\star/2}\right] \leq \mathbb{E}\left[\sum_{m=1}^{M}|\varphi_m - \mu[1]|^{p_\star}\right] = C_{p_\star}\, M.$$

This concludes the proof.

Lemma 2.7.4 *Let V, V', $\{\varphi_m, m \geq 1\}$ and $\{\varphi'_m, m \geq 1\}$ be $\mathbb{R}$-valued random variables defined on $(\Omega, \mathcal{A}, \mathbb{P})$ and $\mathcal{B} \subset \mathcal{A}$ be a σ-field on Ω such that conditionally on $\mathcal{B}$, $\{\varphi_m, m \geq 1\}$ are i.i.d. and $\{\varphi'_m, m \geq 1\}$) are i.i.d. Assume*

(i) the distribution of V admits a density with respect to the Lebesgue measure on $\mathbb{R}$, which is upper bounded by $C_0 > 0$;

(ii) there exists $p_\star > 1$ such that φ_1 and φ'_1 have finite $p_\star$-moments ; set

$$C_{p_\star} := \mathbb{E}\left[|\varphi_1 - \mathbb{E}[\varphi_1 \mid \mathcal{B}]|^{p_\star}\right] \vee \mathbb{E}\left[|\varphi'_1 - \mathbb{E}[\varphi'_1 \mid \mathcal{B}]|^{p_\star}\right].$$

(iii) there exists a constant $C > 0$ such that

$$|V - V'| \leq \frac{C}{M}\left|\sum_{m=1}^{M}(\varphi_m - \mathbb{E}\left[\varphi_1|\mathcal{B}\right])\right| + \frac{1}{M'}\left|\sum_{m=1}^{M'}(\varphi'_m - \mathbb{E}\left[\varphi'_1|\mathcal{B}\right])\right|.$$

Then for any positive integer M,

$$\sup_{\xi\in\mathbb{R}}\mathbb{E}\left[|\mathbf{1}_{V>\xi} - \mathbf{1}_{V'>\xi}|\right] \leq 2^{p_\star}(1 \vee C)^{p_\star}\,(C_0 + c_{p_\star}\, C_{p_\star})\,(M \wedge M')^{-\frac{(p_\star/2)\wedge(p_\star-1)}{1+p_\star}}, \quad (2.7.8)$$

where c_p only depends on p (see its definition in Lemma 2.7.3). If, in addition,

(iv) there exists $C_\infty > 0$ such that for any $\delta > 0$ and any positive integer M,

$$\mathbb{P}\left(\left|\frac{1}{M}\sum_{m=1}^{M}\varphi_m - \mathbb{E}\left[\varphi_1|\mathcal{B}\right]\right| \geq \delta\right) \vee \mathbb{P}\left(\left|\frac{1}{M}\sum_{m=1}^{M}\varphi'_m - \mathbb{E}\left[\varphi'_1|\mathcal{B}\right]\right| \geq \delta\right) \leq e^{-C_\infty M\delta^2},$$

then, for any integers $M, M' \geq 3$,

$$\sup_{\xi\in\mathbb{R}}\mathbb{E}\left[|\mathbf{1}_{V>\xi} - \mathbf{1}_{V'>\xi}|\right] \leq 2\left(1 + \frac{C_0}{\sqrt{2\,C_\infty}}\right)\left(\frac{\log(M \wedge M')}{(M \wedge M')}\right)^{1/2}. \quad (2.7.9)$$

Proof *We have $|\mathbf{1}_{V>\xi} - \mathbf{1}_{V'>\xi}| = \mathbf{1}_{V>\xi\geq V'} + \mathbf{1}_{V'>\xi\geq V}$. Let $\delta > 0$. On the set $\{|V - V'| < \delta\}$, it holds*

$$\{V > \xi \geq V'\} \subset \{V > \xi > V - \delta\}, \qquad \{V' > \xi \geq V\} \subset \{V + \delta > \xi \geq V\},$$

so that

$$\mathbb{E}\left[|\mathbf{1}_{V>\xi} - \mathbf{1}_{V'>\xi}|\mathbf{1}_{|V-V'|<\delta}\right] \leq \mathbb{E}\left[\mathbf{1}_{|V-\xi|<\delta}\mathbf{1}_{|V-V'|<\delta}\right] \leq \mathbb{P}\left(|V - \xi| < \delta\right).$$

By using (i), *it holds*

$$\mathbb{E}[|\mathbf{1}_{V>\xi} - \mathbf{1}_{V'>\xi}|] \leq \mathbb{P}[|V - \xi| < \delta] + \mathbb{P}\left(|V - V'| \geq \delta\right) \leq 2C_0\delta + \mathbb{P}\left(|V - V'| \geq \delta\right). \quad (2.7.10)$$

By (ii)*,* (iii) *and Lemma 2.7.3, and by using* $(x+y)^{p_\star} \le 2^{p_\star-1}(x^{p_\star}+y^{p_\star})$ *for any* $x, y \ge 0$*,*

$$\mathbb{E}\left[|V-V'|^{p_\star}\right] = \mathbb{E}\left[\mathbb{E}\left[|V-V'|^{p_\star} \mid \mathcal{B}\right]\right] \le \frac{2^{p_\star}\,(1\vee C)^{p_\star}\; c_{p_\star}\,C_{p_\star}}{(M\wedge M')^{(p_\star/2)\wedge(p_\star-1)}}.$$

The Chebyshev inequality implies

$$\mathbb{P}(|V-V'|\ge\delta) \le \frac{2^{p_\star}\,(1\vee C)^{p_\star}\; c_{p_\star}\,C_{p_\star}}{\delta^{p_\star}\,(M\wedge M')^{(p_\star/2)\wedge(p_\star-1)}}. \tag{2.7.11}$$

We then obtain (2.7.8) *from* (2.7.10) *and* (2.7.11) *applied to* $\delta \leftarrow (M\wedge M')^{-\widetilde{p}_\star/(1+p_\star)}$ *with* $\widetilde{p}_\star := (p_\star/2)\wedge(p_\star-1)$*.*

Under (iv)*, the second term in* (2.7.10) *is upper bounded by*

$$\exp\left(-C_\infty\,M\,\delta^2/(4C^2)\right) + \exp\left(-C_\infty\,M'\,\delta^2/4\right) \le 2\exp\left(-\frac{C_\infty}{4(1\vee C)^2}\,\delta^2(M\wedge M')\right).$$

Hence we have, by setting $\widetilde{C}_\infty := C_\infty/(4(1\vee C)^2)$ *and* $\overline{M} := M\wedge M'$*,*

$$\mathbb{E}[|\mathbf{1}_{V>\xi} - \mathbf{1}_{V'>\xi}|] \le 2C_0\delta + 2e^{-\widetilde{C}_\infty\,\delta^2\overline{M}} \le \sqrt{2}\frac{C_0}{\sqrt{\widetilde{C}_\infty}}\sqrt{\frac{\ln\overline{M}}{\overline{M}}} + \frac{2}{\sqrt{\overline{M}}},$$

where the last inequality is obtained by choosing $\delta \leftarrow \sqrt{(\log\overline{M})/(2\widetilde{C}_\infty\overline{M})}$*. This concludes the proof since* $\sqrt{\ln\overline{M}} \ge 1$ *for* $\overline{M}\ge 3$*.*

Proof of Lemma 2.3.1. We apply Lemma 2.7.4 with $\mathbb{P}\leftarrow\mathbb{P}_0$, $\mathcal{B}\leftarrow\sigma(Z_0,Z_1)$, $C_i \leftarrow C_i(z)$ for $i\in\{0,p_\star,\infty\}$, $C\leftarrow c_\beta$ and $p_\star\ge 2$. This yields the inequalities (2.3.3) and (2.3.5). Since $|a^+-b^+|\le|a-b|$ and $p_\star\ge1$, we have

$$\mathbb{E}_0\left[\left|(L-\xi)^+-(L'-\xi)^+\right|\right] \le \left(\mathbb{E}_0\left[\left|(L-\xi)^+-(L'-\xi)^+\right|^{p_\star}\right]\right)^{1/p_\star} \le \left(\mathbb{E}_0\left[|L-L'|^{p_\star}\right]\right)^{1/p_\star}.$$

We conclude the proof of (2.3.4) by Lemma 2.7.3.

Proof of Theorem 2.3.1. As in the proof of Theorem 2.2.1, we first establish the results on the sequence $\{\xi_k, k\ge0\}$ by application of Theorem 2.7.1. We then prove the results on the sequence $\{\chi_k, k\ge0\}$ by application of Lemma 2.7.2.

We check the assumptions of Theorem 2.7.1 with $\theta_k\leftarrow\xi_k$ (so that $d=1$), $\mathbb{P}\leftarrow\mathbb{P}_0$, $V_k\leftarrow L_k$, the distribution of $V_\star$ is $\mathrm{Q}(z,\cdot)$, $\mathcal{G}_k\leftarrow\sigma(L_j, j\le k)$ and

$$H(\theta,V)\leftarrow 1-\frac{1}{1-\alpha}\mathbf{1}_{V>\theta}, \qquad h(\theta)\leftarrow 1-\frac{1}{1-\alpha}\mathbb{P}_0(L>\theta).$$

Under H7 and H5, the conditions (i), (ii), (iii) and (v) hold ; the proof is on the same lines as in the proof of Theorem 2.2.1 and is omitted. We establish the condition (vi) (which also implies the condition (iv)) with

$$r_{k+1}\leftarrow\mathbb{E}_0\left[H(\xi_k,L_{k+1})|\mathcal{G}_k\right] - \mathbb{E}_0\left[H(\xi_k,\widetilde{L}_{k+1})|\mathcal{G}_k\right]$$

where

$$\widetilde{L}_{k+1} := \phi^{k+1} + \beta^{k+1}\Psi(Z_1^{k+1}) - \Psi'(z).$$

Note that since the r.v. $(\phi^{k+1}, \beta^{k+1}, Z_1^{k+1})$ are independent from $\mathcal{G}_k$ and, conditionnally to $\mathcal{G}_k$, have the same distribution as the processes (ϕ, β, Z_1), then the distribution of $\widetilde{L}_{k+1}$ given $\mathcal{G}_k$ is $\mathrm{Q}(z, \cdot)$. Hence,

$$\mathbb{E}_0 \left[H(\xi_k, \widetilde{L}_{k+1}) | \mathcal{G}_k \right] = h(\xi_k). \tag{2.7.12}$$

We write

$$r_{k+1} = \mathbb{E}_0 \left[H(\xi_k, L_{k+1}) - H(\xi_k, \widetilde{L}_{k+1}) \middle| \mathcal{G}_k \right] = \frac{1}{1-\alpha} \mathbb{E}_0 \left[\left(\mathbf{1}_{\widetilde{L}_{k+1} > \xi_k} - \mathbf{1}_{L_{k+1} > \xi_k} \right) \middle| \mathcal{G}_k \right]$$

so that, by Lemma 2.3.1, there exists a constant c such that for any $k \geq 1$, $\mathbb{P}_0$-a.s.

$$|r_{k+1}| \leq c \left(M_{k+1} \wedge M'_{k+1} \right)^{-p_\star/(2(1+p_\star))};$$

this implies, by H7, $\sum_{k \geq 1} \gamma_k^{1-\kappa} \mathbb{E}_0[|r_k|^2] < +\infty$, thus proving (vi). This concludes the proof of the results on the sequence $\{\xi_k, k \geq 1\}$.

For the results on the sequence $\{\chi_k, k \geq 1\}$, we check the assumptions of Lemma 2.7.2 with $\theta_k \leftarrow \chi_k$ (so that $d = 1$) and $U_{k+1} \leftarrow \xi_k + (1-\alpha)^{-1}(L_{k+1} - \xi_k)^+$. We write

$$\begin{aligned} U_{k+1} &= \xi_k + (1-\alpha)^{-1} \int (x - \xi_k)^+ \mathrm{Q}(z, \mathrm{d}x) + \widetilde{e}_{k+1} + \widetilde{r}_{k+1} \\ \widetilde{r}_{k+1} &:= (1-\alpha)^{-1} \left((L_{k+1} - \xi_k)^+ - (L'_{k+1} - \xi_k)^+ \right) \\ \widetilde{e}_{k+1} &:= (1-\alpha)^{-1} \left((L'_{k+1} - \xi_k)^+ - \int (x - \xi_k)^+ \mathrm{Q}(z, \mathrm{d}x) \right). \end{aligned} \tag{2.7.13}$$

As in the proof of Theorem 2.2.1, we prove by using H7, H5, that

$$\begin{aligned} &\lim_{k \to \infty} \frac{1}{S_k} \sum_{j=1}^k S_j \gamma_j \xi_j = \xi_\infty \qquad \mathbb{P}_0 - \text{a.s.}, \\ &\lim_{k \to \infty} \frac{1}{S_k} \sum_{j=1}^k S_j \gamma_j \int (x - \xi_j)^+ \mathrm{Q}(z, \mathrm{d}x) = \int (x - \xi_\infty)^+ \mathrm{Q}(z, \mathrm{d}x) \qquad \mathbb{P}_0 - \text{a.s.} \\ &\lim_{k \to +\infty} \frac{1}{S_k} \sum_{j=1}^k S_j \gamma_j \widetilde{e}_{j+1} = 0, \qquad \mathbb{P}_0 - \text{a.s.} \end{aligned}$$

By Lemma 2.3.1 and H7, $\sum_k \gamma_k \mathbb{E}_0 \left[|\widetilde{r}_{j+1}| \right] < \infty$ so that by the Kronecker Lemma,

$$\lim_{k \to +\infty} \frac{1}{S_k} \sum_{j=1}^k S_j \gamma_j \widetilde{r}_{j+1} = 0, \qquad \mathbb{P}_0 - \text{a.s.}$$

By Lemma 2.7.2, we obtain that $\mathbb{P}_0$-a.s., $\lim_k \chi_k$ exists and solves

$$\lim_k \chi_k = \xi_\infty + \frac{1}{1-\alpha} \int (x - \xi_\infty)^+ \mathrm{Q}(z, \mathrm{d}x).$$

2.7.5 A Central Limit Theorem for Stochastic Approximation Algorithms

We recall in this section sufficient conditions for a central limit theorem (CLT) to hold for random variables $\{\theta_k, k \geq 0\}$ defined through a stochastic approximation algorithm : given a deterministic sequence $\{\gamma_k, k \geq 1\}$, a function $h : \mathbb{R}^d \to \mathbb{R}$, $\theta_0 \in \mathbb{R}^d$ and $\mathbb{R}^d$-valued random variables $\{e_k, k \geq 1\}$ and $\{r_k, k \geq 1\}$ defined on $(\Omega, \mathcal{A}, \mathbb{P})$, define for $k \geq 0$,

$$\theta_{k+1} = \theta_k + h(\theta_k) + \gamma_{k+1} e_{k+1} + \gamma_{k+1} r_{k+1}. \tag{2.7.14}$$

Theorem 2.7.2 corresponds to [57, Theorem 2.1.]. It provides sufficient conditions for a CLT along a converging sequence $\{\lim_q \theta_q = \theta_\star\}$ where $\theta_\star \in \mathbb{R}^d$ is fixed (deterministic). On the mean field h and the limit point $\theta_\star$, it is assumed

C 1 *a) The mean field $h : \mathbb{R}^d \to \mathbb{R}^d$ is measurable and twice continuously differentiable in a neighborhood of $\theta_\star$, where $h(\theta_\star) = 0$.*

b) The gradient $\nabla h(\theta_\star)$ is a Hurwitz matrix. Denote by $-\ell$, $(\ell > 0)$, the largest real part of its eigenvalues.

The sequence $\{e_k, k \geq 1\}$ satisfies

C 2 *a) $\{e_k, k \geq 1\}$ is a $\mathcal{G}_k$-adapted $\mathbb{P}$-martingale increment sequence : $\mathbb{E}\left[e_k | \mathfrak{G}_{k-1}\right] = 0$ $\mathbb{Q}$-a.s. for any $k \geq 1$.*

b) For some $C > 0$, there exists $\tau > 0$, such that

$$\sup_k \mathbb{E}\left[|e_{k+1}|^{2+\tau} \mathbf{1}_{|\theta_k - \theta_\star| \leq C}\right] < \infty.$$

c) There exists a symmetric positive definite matrix $D_\star$ and a sequence $\{D_k, k \geq 1\}$ of $\mathbb{R}^d$-valued random variables, such that $\mathbb{P}$-a.s.

$$\mathbb{E}\left[e_{k+1} e_{k+1}^T | \mathcal{G}_k\right] = D_\star + D_k, \qquad \lim_k D_k \mathbf{1}_{\lim_q \theta_q = \theta_\star} = 0.$$

The sequences $\{r_k, k \geq 1\}$ and $\{\gamma_k, k \geq 1\}$ satisfy

C 3 *r_k is $\mathcal{G}_k$-adapted and $\gamma_k^{-1/2} |r_k| \mathbf{1}_{\lim_q \theta_q = \theta_\star} \leq X_k Y_k$ where $\mathbb{P}(\sup_k |X_k| < \infty) = 1$ and and $\lim_k \mathbb{E}[|Y_k|] = 0$.*

C 4 *One of the following condition holds*

a) $\gamma_k \sim \gamma_\star / k$ and $\gamma_\star > 1/(2\ell)$.

b) $\gamma_k \sim \gamma_\star / k^c$ where $c \in (1/2, 1)$.

Theorem 2.7.2 *[57, Theorem 2.1.] Let $\{\theta_k, k \geq 1\}$ be the sequence given by (2.7.14) for some $\theta_0 \in \mathbb{R}^d$. Assume C1, C2, C3 and C4. Let Γ be the positive definite matrix satisfying $\mathbb{Q}$-a.s. on the set $\{\lim_q \theta_q = \theta_\star\}$*

$$\begin{aligned} &\Gamma(Id + 2\gamma_\star \nabla h(\theta_\star)^T) + (Id + 2\gamma_\star \nabla h(\theta_\star))\Gamma = -2\gamma_\star D_\star && \textit{under C4a}, \\ &\Gamma \nabla h(\theta_\star)^T + \nabla h(\theta_\star) \Gamma = -D_\star && \textit{under C4b}. \end{aligned}$$

Then under the conditional probability $\mathbb{Q}(. \mid \lim_q \theta_q = \theta_\star)$ the sequence $\{\gamma_k^{-1/2}(\theta_k - \theta_\star), k \geq 1\}$ converges in distribution to a random variable with the characteristic function

$$\frac{1}{\mathbb{Q}(. \mid \lim_q \theta_q = \theta_\star)} \mathbb{E}_0 \left[\mathbf{1}_{\lim_q \theta_q = \theta_\star} \exp(-\frac{1}{2} t^T \Gamma t)\right].$$

Theorem 2.7.3 corresponds to [57, Theorem 3.2.] ; it provides sufficient conditions for a CLT for the averaged sequence

$$\bar{\theta}_k := \frac{1}{k}\sum_{l=1}^{k}\theta_l,$$

along a converging sequence $\{\lim_q \theta_q = \theta_\star\}$ where $\theta_\star \in \mathbb{R}^d$ is fixed (deterministic). It is established under essentially the same assumptions as in Theorem 2.7.2 that, if C3 is strenghtened into

C 5 *r_k is $\mathcal{G}_k$-adapted and $\gamma_k^{-1/2}|r_k|\mathbf{1}_{\lim_q \theta_q=\theta_\star} \leq X_k Y_k$ where $\mathbb{P}(\sup_k |X_k| < \infty) = 1$ and $\lim_k \mathbb{E}\left[|Y_k|^2\right] = 0$. In addition, $k^{-1/2}\sum_{l=1}^{k} r_l \mathbf{1}_{\lim_q \theta_q=\theta_\star}$ converges to 0 in probability,*

then :

Theorem 2.7.3 *[57, Theorem 3.2.] Assume C 1, C2, C4b, and C5. Then for any $t \in \mathbb{R}^d$,*

$$\lim_k \mathbb{E}\left[\exp(i\sqrt{k}t^T(\bar{\theta}_k - \theta_\star))|\lim_q \theta_q = \theta_\star\right] = \exp\left(-\frac{1}{2}t^T\nabla h(\theta_\star)^{-1}D_\star(\nabla h(\theta_\star)^{-1})^T t\right),$$

where $D_\star$ is introduced in C2.

2.7.6 Proofs of the Results of Section 2.3.2

Throughout this section, set $\theta := (\xi, \chi)$, and

$$h(\theta) := -\begin{bmatrix} 1 - \frac{1}{1-\alpha}\mathbb{P}_0\left(L > \xi\right) \\ \chi - \xi - \frac{1}{1-\alpha}\mathbb{E}_0\left[(L-\xi)^+\right]\end{bmatrix}, \quad H(\theta, X) := -\begin{bmatrix} 1 - \frac{1}{1-\alpha}\mathbf{1}_{X>\xi} \\ \chi - \xi - \frac{1}{1-\alpha}\left(X-\xi\right)^+\end{bmatrix}. \tag{2.7.15}$$

We start with a preliminary lemma.

Lemma 2.7.5 *Assume H9. Then*

$$\mathbb{E}_0\left[\left(H(\theta,L) - h(\theta)\right)\left(H(\theta,L) - h(\theta)\right)^T\right] = D_\star + D(\theta), \tag{2.7.16}$$

where

$$D_\star := \frac{1}{(1-\alpha)^2}\begin{bmatrix} \alpha(1-\alpha) & \alpha\,\mathbb{E}_0\left[(L-\xi_\star)^+\right] \\ \alpha\,\mathbb{E}_0\left[(L-\xi_\star)^+\right] & \mathbb{V}\mathrm{ar}_0\left[(L-\xi_\star)^+\right]\end{bmatrix} \tag{2.7.17}$$

and $\lim_k D(\theta_k)\mathbf{1}_{\lim_q \theta_q=\theta_\star} = 0$ $\mathbb{P}_0$-a.s

Proof *By H9, $L \sim \mathrm{Q}(z,\cdot)$ under $\mathbb{P}_0$ so $\mathbb{E}_0\left[H(\theta,L)\right] = h(\theta)$. We have*

$$\mathbb{E}_0\left[\left(H(\theta,L) - h(\theta)\right)\left(H(\theta,L) - h(\theta)\right)^T\right] = \mathbb{E}_0\left[H(\theta,L)H(\theta,L)^T\right] - h(\theta)h(\theta)^T.$$

Denote by $\mathcal{H}(\theta)$ the first 2×2 symmetric matrix on the RHS. Then

$$\mathcal{H}(\theta)_{1,1} = 1 + \frac{\mathbb{P}_0\left(L > \xi\right)}{(1-\alpha)^2} - \frac{2}{1-\alpha}\mathbb{P}_0\left(L > \xi\right)$$

$$\mathcal{H}(\theta)_{1,2} = \chi - \xi - \frac{1}{1-\alpha}\mathbb{E}_0\left[(L-\xi)^+\right] - \frac{1}{1-\alpha}\mathbb{P}_0\left(L > \xi\right)(\chi - \xi) + \frac{1}{(1-\alpha)^2}\mathbb{E}_0\left[(L-\xi)^+\right]$$

$$\mathcal{H}(\theta)_{2,2} = (\chi-\xi)^2 + \frac{1}{(1-\alpha)^2}\mathbb{E}_0\left[\left((L-\xi)^+\right)^2\right] - \frac{2}{1-\alpha}(\chi-\xi)\mathbb{E}_0\left[(L-\xi)^+\right].$$

Under H9, $\theta \mapsto h(\theta)$ and $\theta \mapsto \mathcal{H}(\theta)$ are continuous at $\theta_\star$. This implies (2.7.16)*, where the expression* (2.7.17) *for*

$$D_\star = \mathcal{H}(\theta_\star) - h(\theta_\star)h(\theta_\star)^T.$$

follows by using that $\theta_\star$ satisfies (2.2.6) *(which implies that $h(\theta_\star) = 0$).*

Proof of Theorem 2.3.2 The proof consists in applying Theorem 2.7.2. We check its assumptions with $\mathbb{Q} \leftarrow \mathbb{P}_0$, $\theta_k \leftarrow (\xi_k, \chi_k)$, $\theta_\star \leftarrow (\xi_\star, \chi_\star)$, the function h given by (2.7.15). The random variables e_k, r_k are set equal to

$$e_{k+1} \leftarrow H(\theta_k, \widetilde{L}_{k+1}) - h(\theta_k) \qquad r_{k+1} \leftarrow H(\theta_k, L_{k+1}) - H(\theta_k, L'_{k+1})$$

where h and H are given by (2.7.15) and

$$\widetilde{L}_{k+1} := \phi^{k+1} + \beta^{k+1}\Psi(Z_1^{k+1}) - \Psi'(z).$$

With these definitions, note that Algorithm 2 updates the parameter θ_{k+1} by

$$\theta_{k+1} = \theta_k + \gamma_{k+1} H(\theta_k, L_{k+1}).$$

Since $\theta_\star$ satisfies (2.2.6), we have $h(\theta_\star) = 0$. By H9, the function h is twice continuously differentiable in a neighborhood of $\theta_\star$ and the gradient is given by

$$\nabla h(\theta_\star) = -\frac{1}{1-\alpha}\begin{bmatrix} f(z,\xi_\star) & 0 \\ -(1-\alpha) + \mathbb{P}_0(L > \xi_\star) & 1-\alpha \end{bmatrix} = -\begin{bmatrix} \frac{1}{1-\alpha} f(z,\xi_\star) & 0 \\ 0 & 1 \end{bmatrix} \tag{2.7.18}$$

where we used (2.2.6) in the last equality. Hence, by H9, the condition C1 is verified.

Set $\mathcal{G}_k := \sigma\left(L_j, j \le k\right)$; note that $h(\theta_k) = \mathbb{E}_0\left[H(\theta_k, \widetilde{L}_{k+1}) | \mathcal{G}_k\right]$ – see (2.7.12) in the proof of Theorem 2.3.1. Hence, $\{e_k, k \ge 1\}$ is a $\mathcal{G}_k$-adapted $\mathbb{P}_0$-martingale increment sequence. Since $\theta \mapsto h(\theta)$ is continuous at $\theta_\star$ and $\theta_\star$ is fixed (deterministic), for fixed $C > 0$, there exists a constant C' such that $\mathbb{P}_0$-a.s., $\sup_k |h(\theta_k)| \mathbf{1}_{|\theta_k - \theta_\star| \le C} \le C'$. In addition

$$|H(\theta, X)| \le (1 + 1/(1-\alpha)) + 2|\theta| + (1-\alpha)^{-1}(X - \xi)^+.$$

By H9, since $\widetilde{L}_{k+1}$ has the same distribution as L under $\mathbb{P}_0$, there exists a constant C'' (depending upon C) such that

$$\sup_k \mathbb{E}_0\left[|H(\theta_k, \widetilde{L}_{k+1})|^{2+\tau_\star} \mathbf{1}_{|\theta_k - \theta_\star| \le C}\right] \le C''.$$

Hence, the conditions C2a-b are verified. The condition C2c follows from Lemma 2.7.5. We write

$$\begin{aligned} \mathbb{E}_0\left[|r_{k+1}|\right] &= \mathbb{E}_0\left[\mathbb{E}_0\left[|r_{k+1}|\, |\mathcal{G}_k\right]\right] \\ &\le \frac{1}{1-\alpha}\left(\sup_{\xi \in \mathbb{R}} \mathbb{E}_0\left[\left|\mathbf{1}_{L_{k+1} > \xi} - \mathbf{1}_{\widetilde{L}_{k+1} > \xi}\right|\right] + \sup_{\xi \in \mathbb{R}} \mathbb{E}_0\left[\left|(L_{k+1} - \xi)^+ - \left(\widetilde{L}_{k+1} - \xi\right)^+\right|\right]\right). \end{aligned}$$

By Lemma 2.3.1, under H9, the LHS is upper bounded by

$$O\left((M_k \wedge M'_k)^{-p_\star/(2(1+p_\star))} + (M_k \wedge M'_k)^{-1/2}\right) = O\left((M_k \wedge M'_k)^{-p_\star/(2(1+p_\star))}\right).$$

Hence, by H10, the condition C3 is verified. Finally, the condition C4 holds by H10 and (2.7.18). This concludes the proof of the theorem.

Proof of Theorem 2.3.3 The proof consist in an application of Theorem 2.7.3. We use the same notations as in the proof of Theorem 2.3.2 ; it was already proved that C1 and C2 hold. We check C5 : we have

$$\mathbb{E}_0\left[|r_{k+1}|^2\right] \leq \frac{1}{(1-\alpha)^2}\sup_{\xi\in\mathbb{R}}\left(\mathbb{E}_0\left[\left|\mathbf{1}_{L_{k+1}>\xi}-\mathbf{1}_{\widetilde{L}_{k+1}>\xi}\right|\right]+\mathbb{E}_0\left[\left|(L_{k+1}-\xi)^+-(\widetilde{L}_{k+1}-\xi)^+\right|^2\right]\right).$$

By Lemma 2.3.1, there exists a constant C such that for any $k \geq 0$,

$$\mathbb{E}_0\left[|r_{k+1}|^2\right] \leq \frac{C}{(M_k \wedge M'_k)^{p_\star/(2(1+p_\star))}}.$$

Therefore, the condition on $\gamma_k^{-1/2} r_k$ is satisfied by (2.3.8). In addition, by Lemma 2.3.1 again, there exists a constant C' such that for any $\delta > 0$,

$$\mathbb{P}_0\left(k^{-1/2}\left|\sum_{l=1}^{k} r_l\right| > \delta\right) \leq k^{-1/2}\delta^{-1}\mathbb{E}_0\left[\left|\sum_{l=1}^{k} r_l\right|\right] \leq C' k^{-1/2}\delta^{-1}\left(\sum_{l=1}^{k}\frac{1}{(M_l \wedge M'_l)^{p_\star/(2(1+p_\star))}}\right).$$

Therefore, the condition C5 holds by (2.3.8).

2.7.7 Sensitivities of Value-at-Risk and Expected Shortfall to Perturbations of the Input Distribution

We develop in this section some estimates relative to the perturbation of the value-at-risk and expected shortfall that arise when we use different distributions for the underlying loss variable Z. We use the notation $\mathbb{V}a\mathbb{R}^\alpha(Z)$ and $\mathbb{ES}^\alpha(Z)$ for the $\mathbb{Q}$ value-at-risk and expected shortfall of Z

$$\mathbb{Q}(Z > \mathbb{V}a\mathbb{R}^\alpha(Z)) := 1-\alpha, \qquad \mathbb{ES}^\alpha(Z) := \frac{1}{1-\alpha}\int_\alpha^1 \mathbb{V}a\mathbb{R}^a(Z)\,\mathrm{d}a,$$

where $\mathbb{V}a\mathbb{R}^\alpha(Z)$ defined on the left is the infimum of such values.

Definition 2.7.1 *The Kolmogorov distance $d_{\mathrm{kol}}(X,Y)$ between two scalar random variables X and Y is the* sup *norm between their cumulative distribution functions :*

$$d_{\mathrm{kol}}(X,Y) := \sup_{\xi\in\mathbb{R}}|\mathbb{Q}[X \leq \xi] - \mathbb{Q}[Y \leq \xi]|.$$

We show that if X, Y are integrable scalar random variables with a continuous density, then for any $\alpha > 0$ fixed, the difference $|[\mathbb{V}a\mathbb{R}^\alpha(X), \mathbb{ES}^\alpha(X)] - [\mathbb{V}a\mathbb{R}^\alpha(Y), \mathbb{ES}^\alpha(Y)]|$ is bounded, up to a multiplicative constant depending of α and the density of X, by the L^1 and the Kolmogorov distances between X and Y.

Our first proposition has to do with the relationship between the Kolmogorov distance and the behavior of $\mathbb{V}a\mathbb{R}^\beta(\cdot)$ as a function of β.

Lemma 2.7.6 *Let X and Y be scalar random variables having a continuous cumulative distribution function. Then for any $\alpha \in (0,1)$ and every $\mathbb{V}a\mathbb{R}^\alpha(Y)$ we have*

$$\mathbb{V}a\mathbb{R}^{\alpha - d_{\mathrm{kol}}(X,Y)}(X) \leq \mathbb{V}a\mathbb{R}^\alpha(Y) \leq \mathbb{V}a\mathbb{R}^{\alpha + d_{\mathrm{kol}}(X,Y)}(X). \qquad (2.7.19)$$

for some elements from the respective $\mathbb{V}a\mathbb{R}^\alpha(X)$ sets and with the convention $\mathbb{V}a\mathbb{R}^\beta = -\infty$ (respectively $\mathbb{V}a\mathbb{R}^\beta = \infty$) if $\beta < 0$ (respectively $\beta > 1$).

If X and Y are also integrable then

$$|\mathbb{ES}^\alpha(X) - \mathbb{ES}^\alpha(Y)| \leq \frac{1}{1-\alpha}\mathbb{E}|X-Y|. \qquad (2.7.20)$$

Proof *Let $\alpha \in (0,1)$ be given, and let $d := d_{\mathrm{kol}}(X,Y)$. From the definition of the Kolmogorov distance it follows that for every $\xi \in \mathbb{R}$*

$$\mathbb{Q}[X \leq \xi] - d \leq \mathbb{Q}[Y \leq \xi] \leq \mathbb{Q}[X \leq \xi] + d,$$

so that for every ξ^α such that $\mathbb{Q}[Y \leq \xi^\alpha] = \alpha$ (i.e. for every $\mathbb{V}a\mathbb{R}^\alpha$ element of Y) we have

$$|\mathbb{Q}[X \leq \xi^\alpha] - \alpha| \leq d,$$

thus showing that $\mathbb{V}a\mathbb{R}^{\alpha-d}(X) \leq \xi^\alpha \leq \mathbb{V}a\mathbb{R}^{\alpha+d}(X)$ for some elements in the respective $\mathbb{V}a\mathbb{R}$ sets.

For the second equality note that, by the characterization in [28, Section 2.1]-[109, Theorem 1] of $\mathbb{ES}^\alpha(Z)$, for $Z = X$ or $Z = Y$ (under the assumptions of continuous c.d.f.),

$$\mathbb{ES}^\alpha(Z) = \inf_x \left(x + \frac{1}{1-\alpha}\mathbb{E}\left[(Z-x)^+\right]\right). \tag{2.7.21}$$

Now consider the function

$$G(x,z) := x + \frac{1}{1-\alpha}(z-x)^+$$

and note that, for fixed x, the function $G(x,\cdot)$ is a uniformly Lipschitz function of z with Lipschitz constant $1/(1-\alpha)$. This implies in particular, by taking $Z = X$ and $Z = Y$, that for every x

$$\mathbb{E}\left[G(x,X)\right] - \mathbb{E}\left[G(x,Y)\right] | \leq \frac{1}{1-\alpha}\mathbb{E}|X-Y|. \tag{2.7.22}$$

Taking the inf in x in the above and using (2.7.21), we get (2.7.20) as desired.

Inspired from [26] , we develop further estimates on the Kolmogorov distance between X and Y that might depend on higher moments for the difference between these random variables. We apply these estimates to the error analysis of Algorithm 3, in which the bias due to fixing an approximation procedure for the samplings of Ψ has to be controlled in order to have useful criteria for the choice of the parameters of the algorithm.

Corollary 2.7.1 *Assume that the scalar random variable X has a c.d.f. F which is continuously differentiable and strictly increasing in a neighborhood of $\mathbb{V}a\mathbb{R}^\alpha(X)$, let $f := dF/d\lambda$ be the respective density (where it exists), and let δ be such that the inverse F^{-1} of F exists in an $\delta-$neighborhood of α. Then for any scalar random variable Y and any $0 < r, s < \delta$, the condition*

$$\mathbb{V}a\mathbb{R}^{\alpha-r}(X) \leq \mathbb{V}a\mathbb{R}^\alpha(Y) \leq \mathbb{V}a\mathbb{R}^{\alpha+s}(X)$$

implies that

$$|\mathbb{V}a\mathbb{R}^\alpha(X) - \mathbb{V}a\mathbb{R}^\alpha(Y)| \leq \sup_{x\in[\alpha-r,\alpha+s]} |f(F^{-1}(x))|^{-1} d_{\mathrm{kol}}(X,Y). \tag{2.7.23}$$

Proof *This follows from the fact that, under the given hypotheses*

$$\mathbb{V}a\mathbb{R}^{\beta}(X) = F^{-1}(\beta)$$

whenever $|\beta - \alpha| < \delta$. *The hypotheses on* r *and* s *imply now that*

$$\begin{aligned} |\mathbb{V}a\mathbb{R}^{\alpha}(X) - \mathbb{V}a\mathbb{R}^{\alpha}(Y)| &\leq \max\Big(\mathbb{V}a\mathbb{R}^{\alpha}(X) - \mathbb{V}a\mathbb{R}^{\alpha-r}(X), \mathbb{V}a\mathbb{R}^{\alpha+s}(X) - \mathbb{V}a\mathbb{R}^{\alpha}(X)\Big) \qquad (2.7.24) \\ &= \max\Big(F^{-1}(\alpha) - F^{-1}(\alpha - r), F^{-1}(\alpha + s) - F^{-1}(\alpha)\Big). \end{aligned}$$

It follows from the mean value theorem and the inverse function theorem that

$$|\mathbb{V}a\mathbb{R}^{\alpha}(X) - \mathbb{V}a\mathbb{R}^{\alpha}(Y)| \leq \sup_{x\in[\alpha-r,\alpha+s]} |f(F^{-1}(x))|^{-1} \max(r, s)$$

which implies the desired conclusion if $\max(r, s) < d_{\text{kol}}(X, Y)$.

If this is not the case, for instance if $r \leq d_{\text{kol}}(X, Y) < s$, *a similar argument using* (2.7.19) *(available since* X *has a continuous c.d.f.) to replace the second term in the maximum* (2.7.24) *gives*

$$\begin{aligned} |\mathbb{V}a\mathbb{R}^{\alpha}(X) - \mathbb{V}a\mathbb{R}^{\alpha}(Y)| &\leq \sup_{x\in[\alpha-r,\alpha+d_{\text{kol}}(X,Y)]} |f(F^{-1}(x))|^{-1}\max(r, d_{\text{kol}}(X, Y)) \\ &\leq \sup_{x\in[\alpha-r,\alpha+s]} |f(F^{-1}(x))|^{-1} d_{\text{kol}}(X, Y). \end{aligned}$$

The other cases are treated similarly.

In order to pass to controls that depend only on the L^1 distance, we present now two estimates of $d_{\text{kol}}(X, Y)$ that are related to the actual difference between X and Y. These will be combined to estimate the expected error induced by the application of the stochastic approximation procedure to the sequence of samplings produced via regression.

Lemma 2.7.7 *Assume that the scalar random variable* X *admits a density which is bounded by* C_0. *Then for any scalar random variable* Y *and any* $\delta > 0$

$$d_{\text{kol}}(X, Y) \leq 2C_0\delta + \mathbb{P}\left[|X - Y| > \delta\right]. \qquad (2.7.25)$$

Proof *The following argument was presented already in the proof of Lemma 2.7.4, thus we give here a summarized version : for* $\delta > 0$ *given and any* $\xi \in \mathbb{R}$, *we have*

$$\begin{aligned} |\mathbb{P}\left[X \leq \xi\right] - \mathbb{P}\left[Y \leq \xi\right]| &\leq \mathbb{E}\left[|\mathbb{1}_{[X\leq\xi]} - \mathbb{1}_{[Y\leq\xi]}|\right] \\ &= \mathbb{E}\left[\mathbb{1}_{[X\leq\xi<Y]} + \mathbb{1}_{[Y\leq\xi<X]}\right] \\ &\leq \mathbb{E}\left[\mathbb{1}_{[-\delta+X\leq\xi\leq X+\delta]}\right] + \mathbb{E}\left[\mathbb{1}_{[|X-Y|>\delta]}\right] \\ &\leq 2C_0\delta + \mathbb{P}\left[|X - Y| > \delta\right], \end{aligned}$$

using the hypothesis.

Corollary 2.7.2 *Assume that the scalar random variable* X *admits a density which is bounded by* C_0. *Then for any scalar random variable* Y *and any* $p > 0$ *we have*

$$d_{\text{kol}}(X, Y) \leq (2C_0 + 1)(\mathbb{E}|X - Y|^p)^{1/(1+p)}.$$

Proof *For the case in which* $\mathbb{E}|X - Y|^p = +\infty$ *the conclusion is trivially true. For the* $p-$*integrable case, take* $\delta = (\mathbb{E}|X - Y|^p)^{1/(1+p)}$ *in equation* (2.7.25) *and apply Markov's inequality.*

2.7.8 A Nonasymptotic Estimate for Regressions

The following result is used to control the error due to the introduction of a regression procedure in Algorithm 3 :

Theorem 2.7.4 ([84, Theorem 11.5]) *Let (X, Y) be a random vector in $\mathbb{R}^d \times \mathbb{R}$, let $\mathcal{F}$ be a* pointwise measurable *set of functions $f : \mathbb{R}^d \to \mathbb{R}$, with finite Vapnik-Chervonenkis dimension $\mathbf{VC}_{\mathcal{F}} \geq 1$. Assume that the random variable Y is bounded by $B > 0$. If $\mathcal{D}_n = ((X_k, Y_k))_{k=1}^n$ is any vector of independent copies of (X, Y) and if we define the random function $f_{\mathcal{D}_n}$ by*

$$f_{\mathcal{D}_n} := \widehat{f}_{\mathcal{D}_n} \mathbb{1}_{|\widehat{f}_{\mathcal{D}_n}| \leq B} + B \mathbb{1}_{\widehat{f}_{\mathcal{D}_n} > B} - B \mathbb{1}_{\widehat{f}_{\mathcal{D}_n} < -B}$$

where

$$\widehat{f}_{\mathcal{D}_n} := \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{k=1}^{n} |f(X_k) - Y_k|^2,$$

then there exists a universal constant $C_\star$ such that for any copy (X', Y') of (X, Y) independent of $\mathcal{D}_n$,

$$\mathbb{E}\left[|f_{\mathcal{D}_n}(X') - \mathbb{E}[Y' \mid X']|^2\right] \leq C_\star B^2 \mathbf{VC}_{\mathcal{F}} \frac{(1 + \ln(n))}{n} + 2 \inf_{f \in \mathcal{F}} \mathbb{E}\left[|f(X) - \mathbb{E}[Y \mid X]|^2\right]. \tag{2.7.26}$$

Remark 2.7.1 *It may be useful to recall the meaning of* (2.7.26) *: if $\mu_X = \mathbb{Q}X^{-1}$ is the law of X, then for any $\mathcal{D}_n-$measurable nonnegative (random) function $g = g_{\mathcal{D}_n}$ and any copy X' of X independent of $\mathcal{D}_n$,*

$$\mathbb{E}\left[g_{\mathcal{D}_n}(X') \mid \mathcal{D}_n\right] = \int_{\mathbb{R}^d} g_{\mathcal{D}_n}(x) \mathrm{d}\mu_X(x). \tag{2.7.27}$$

Now, if m is a (deterministic) function with the property that

$$m(X) = \mathbb{E}[Y \mid X], \qquad \mathbb{Q} - a.s.$$

(and therefore $m(X') = \mathbb{E}[Y' \mid X']$ because $(X, Y) \sim (X', Y')$), and if we apply (2.7.27) *to $g_{\mathcal{D}_n} := |f_{\mathcal{D}_n} - m|^2$, we get that*

$$\mathbb{E}\left[|f_{\mathcal{D}_n}(X') - \mathbb{E}[Y' \mid X']|^2 | \mathcal{D}_n\right] \geq \inf_{f \in \mathcal{F}} \mathbb{E}\left[|f \circ X' - m \circ X'|^2\right] = \inf_{f \in \mathcal{F}} \mathbb{E}\left[|f(X) - \mathbb{E}[Y \mid X]|^2\right],$$

and therefore (2.7.26) *tells us that, up to a factor of 2 (which can be improved by looking carefully at the proofs), the accuracy of $f_{\mathcal{D}_n}$ as a predictor constructed from $\mathcal{F}$ of Y as a function of X deviates from the optimal* L^2-*accuracy*

$$\inf_{f \in \mathcal{F}} \mathbb{E}\left[|f(X) - \mathbb{E}[Y \mid X]|^2\right]$$

for no more than

$$C B^2 \mathbf{VC}_{\mathcal{F}} \frac{(1 + \ln(n))}{n}$$

units, on $\mathrm{L}^2_{\mathbb{Q}}-$ *expectation.*

About estimating the Vapnik-Chervonenkis dimension of a neural network, see for instance [22] . When $\mathcal{F}$ is a vector space, we can replace the factor $\mathbf{VC}_{\mathcal{F}}$ by the dimension of the vector space plus one.

Chapitre 3

XVA Principles, Nested Monte Carlo Strategies, and GPU Optimizations

This chapter is based on Abbas-Turki, Crépey and Diallo [2].

3.1 Introduction

Since the 2008 financial crisis, investment banks charge to their clients, in the form of rebates with respect to the counterparty-risk-free value of financial derivatives, various add-ons meant to account for counterparty risk and its capital and funding implications. These add-ons are dubbed XVAs, where VA stands for valuation adjustment and X is a catch-all letter to be replaced by C for credit, D for debt, F for funding, M for margin, or K for capital.

XVAs greatly complicate the derivative pricing equations by making them global, nonlinear, and entity-dependent. However, in order to avoid or defer major IT changes, many banks (but not all) tend to content themselves with the following exposure-based approach (see e.g. [39]) : First, they compute the mark-to-market cube of the counterparty-risk free valuation of all their contracts in any scenario and future time point. Then they integrate in time the ensuing expected positive exposure (EPE) profile against the hazard function of the client implied from its credit default swap (CDS) CDS curve. Similar approaches are applied to DVA and FVA computations (using the bank own CDS or funding curves instead of the client CDS curve in the CVA case).

However, exposure-based approaches are purely static, whereas a dynamic perspective is required for (at least rigorous dynamic) XVA hedging purposes and for properly accounting for the feedback effects between different XVAs (e.g. from the CVA into the FVA). Moreover, an exposure-based XVA approach primarily assumes independence between the market and credit sides of the problem : Beyond more or less elaborate patches such as the ones proposed in [104], [78], [88], or [79], it is hard to extend rigorously to wrong-way risk, which is the risk of adverse dependence between the credit risk of the counterparty and the underlying market exposure. Last but not least, an exposure-based XVA approach comes without error control, at least whenever the exposure is computed by global regression involving unconditional approximations. Instead, in this paper, we explore a full simulation, nested Monte Carlo XVA computational approach granting a $\mathrm{O}(M_{(0)}^{-\frac{1}{2}})$ mean square error, where $M_{(0)}$ is the outer number of trajectories, optimally implemented on graphic programming units (GPUs).

[10] advocate a supercomputer XVA implementation, whereby risk factors are simulated forward, whereas the backward pricing task is performed by fast matrix exponentiation. Although extremely fast and accurate whenever applicable, this approach is restricted to models written as piecewise time homogeneous Markov chains (for applicability of the matrix exponentiation formula) with at most three factors

(unless advanced techniques are used for circumventing the curse of dimensionality), Malliavin calculus can be used for extending such an approach to space continuous models (see [1]), but the curse of dimensionality issue remains.

The **outline** of the paper is as follows. Section 4.3 sets the XVA stage. Section 3.3 presents the multi-layered dependence between the different XVA metrics and discusses their nested Monte Carlo (NMC) implementation from a high-level perspective. Section 3.4 illustrates the above by three case studies. Section 3.5 discusses future perspectives. Sections A through D detail the related GPU programming optimization techniques.

The **contributions** of this work are the following : (a) from a high-level perspective, the conceptual view of Sect. 3.3 on the dependence between the XVA metrics, and its algorithmic counterpart in the form of the nested Monte Carlo Algorithm 5 ; (b) from a technical point of view, the XVA NMC GPU programming optimization techniques detailed in the appendix, including the innovative Algorithm 6 for efficient value-at-risk and expected shortfall computations ; and (c) the proof of concept, by the numerical case studies of Sect. 3.4, that nested Monte Carlo XVA computations are within reach if GPU power is properly used (cf. Table 3.9).

3.2 XVA Guidelines

In this section we recall the XVA principles of [12], to which we refer the reader for more details. See also the companion papers by [11] and [24] for applications in respective bilateral and centrally cleared trading setups. For other XVA frameworks, see for instance [36] or [33] (without KVA) or, with a KVA meant as an additional contra-asset like the CVA and the FVA (as opposed to a risk premium in our case), [69], [72], or [53].

We consider a pricing stochastic basis $(\Omega, \mathbb{F}, \mathbb{P})$, for a reference market filtration (ignoring the default of the bank itself) $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{R}_+}$ and a risk-neutral pricing measure $\mathbb{P}$ calibrated to vanilla market quotes, such that all the processes of interest are $\mathbb{F}$ adapted and all the random times of interest are $\mathbb{F}$ stopping times. This holds at least after so-called reduction of all the data to $\mathbb{F}$ starting from a larger filtration $\mathbb{G}$ including the default of the bank itself as a stopping time, supposing immersion from $\mathbb{F}$ into $\mathbb{G}$ for simplicity in this work.

Remark 3.2.1 *[12] explicitly introduce the larger filtration $\mathbb{G}$ and show how to deal with the XVA equations, which are natively stated in $\mathbb{G}$, by reduction to $\mathbb{F}$ (denoting the reduced data with a $\cdot'$). Here we directly state the reduced equations in $\mathbb{F}$ (and we do not use the $\cdot'$ notation).* ∎

The $\mathbb{P}$ expectation and $(\mathcal{F}_t, \mathbb{P})$ conditional expectation are denoted by $\mathbb{E}$ and $\mathbb{E}_t$. We denote by r an $\mathbb{F}$ progressive OIS (overnight indexed swap) rate process, which is together the best market proxy for a risk-free rate and the reference rate for the remuneration of the collateral. We write $\beta = e^{-\int_0^{\cdot} r_s ds}$ for the corresponding risk-neutral discount factor.

We consider a bank trading, in a bilateral and/or centrally cleared setup, with risky counterparties. The bank is also default prone, with default intensity γ and recovery rate R. We denote by T an upper bound on the maturity of all claims in the bank portfolio, also accounting for the time of liquidating the position between the bank and any of its clients in case of default.

By mark-to-market of a contract (or portfolio), we mean the (trade additive) risk-neutral conditional expectation of its future discounted promised cash flows D, ignoring counterparty risk and its capital and funding implications, i.e. without any XVAs. In particular, we denote by MtM the mark-to-market of the overall derivative client portfolio of the bank.

3.2.1 Counterparty Exposure Cash Flows

To mitigate counterparty risk, the bank and its counterparties post variation and initial margin as well as, in a centrally cleared setup, default fund contributions. The variation margin (VM) of each party tracks the mark-to-market of their portfolio at variation margin call times, as long as both parties do not default. However, there is a liquidation period, usually a few days, between the default of a party and the liquidation of its portfolio. Even in the case of a perfectly variation-margined portfolio, the gap risk of slippage of the mark-to-market of the portfolio of a defaulted party and of unpaid contractual cash flows during its liquidation period justifies the need for initial margin (IM). The IM of each party is dynamically set as a risk measure, such as value-at-risk ($\mathbb{V}\mathrm{a}\mathbb{R}$) at some confidence level a_{im}, of their loss-and-profit at the time horizon of the liquidation period (sensitivity $\mathbb{V}\mathrm{a}\mathbb{R}$ in bilateral SIMM, which is the standard initial margin model for non-cleared derivatives, and historical $\mathbb{V}\mathrm{a}\mathbb{R}$ for centrally cleared trading). In case a party defaults, its IM provides a buffer to absorb the losses that may arise on the portfolio from adverse scenarios, exacerbated by wrong-way risk, during the liquidation period.

On top of the variation and initial margins that are used in bilateral transactions, a central counterparty (CCP) deals with extreme and systemic risk on a mutualization basis, through an additional layer of protection, called the default fund (DF), which is pooled between the clearing members. The default fund is used when the losses exceed the sum between the VM and the IM of the defaulted member. The default fund contribution of the defaulted member is used first. If it does not suffice, the default fund contributions of the other clearing members are used in turn. Under the current European EMIR regulation, the Cover 2 rule requires to size the default fund as, at least, the maximum of the largest exposure and of the sum of the second and third largest exposures of the CCP to its clearing members, updated periodically (e.g. monthly) based on “extreme but plausible” scenarios. The corresponding amount is allocated between the clearing members, typically proportionally to their losses over IM (or to their IM itself).

3.2.2 Funding Cash Flows

Variation margin typically consists of cash that is re-hypothecable, meaning that received VM can be used for funding purposes and is remunerated OIS by the receiving party. Initial margin, as well as default fund contributions in a CCP setup, typically consist of liquid assets deposited in a segregated account, such as government bonds, which pay coupons or otherwise accrue in value. We assume that the bank can invest at the OIS rate r and obtain unsecured funding for borrowing VM at the rate $(r+\lambda)$, where the unsecured funding spread $\lambda = (1-R)\gamma$ can be interpreted as an instantaneous CDS spread of the bank. Initial margin is funded separately from variation margin (see the third paragraph of Section 3.2 in [11]) at a blended spread $\bar{\lambda}$ that depends on the IM funding policy of the bank (see Section 4.3 in [11]).

3.2.3 Cost of Capital Pricing Approach in Incomplete Counterparty Credit Risk Markets

In theory, a bank may want to setup an XVA hedge. But, as (especially own) jump-to-default exposures are hard to hedge in pratice, such a hedge can only be very imperfect. In the context of XVA computations, we assume a perfectly collateralized back to back market hedge of its client portfolio by the bank (i.e. the bank posts MtM as variation margin on its hedge), but we conservatively assume no XVA hedge.

To deal with the corresponding market incompleteness issue, we follow a cost of capital XVA pricing approach, in two steps. First, the so-called contra-assets are valued as the expected costs of the counterparty default losses and risky funding expenses. Second, on top of these expected costs, a KVA risk premium (capital valuation adjustment) is computed as the cost of a sustainable remuneration of the shareholder capital at risk which is earmarked by the bank to absorb its exceptional (beyond expected) losses.

More precisely, the contra-asset value process (CA) corresponds to the expected discounted future counterparty default losses and funding expenditures. Incremental CA amounts are charged by the bank to its clients at every new deal and put in a reserve capital account, which is then depleted by counterparty default losses and funding expenditures as they occur.

In addition, bank shareholders require a remuneration at some hurdle rate h, commonly estimated by financial economists in a range varying from 6% to 13%, for the risk on their capital. Accordingly, an incremental risk margin (or KVA) is sourced from clients at every new trade in view of being gradually distributed to bank shareholders as remuneration for their capital at risk at rate h as time goes on.

Remark 3.2.2 *In practice, target return on equities are fixed by the management of the bank every year. In the context of our XVA computations, we take h as an exogenous constant for simplicity.*

Cost of capital calculations involve projections over decades in the future. The historical probability measure is hardly estimable on such time frames. As a consequence, we do all our price and risk computations under the risk-neutral measure $\mathbb{P}$.

The uncertainty on the hurdle rate or on the historical probability measure are left to model risk. ∎

3.2.4 Contra-Assets Valuation

We work under the modeling assumption that every bank account is continuously reset to its theoretical target value, any discrepancy between the two being instantaneously realized as loss or earning by the bank.

In particular, the reserve capital account of the bank is continuously reset to its theoretical target CA level so that, much like with futures, the trading position of the bank is reset to zero at all times, but it generates a trading loss-and-profit process L. Our equation for the contra-assets value process CA is then derived from a risk-neutral martingale condition on the trading loss process L of the bank, along with a terminal condition $\mathrm{CA}_T = 0$. This martingale condition on L corresponds to a bank shareholder no arbitrage condition. It results in

$$\mathrm{CA} = \mathrm{CVA} + \mathrm{FVA} + \mathrm{MVA}, \tag{3.2.1}$$

for setup-dependent, but always nonnegative, CVA, FVA, and MVA processes. The FVA corresponds to the cost of funding cash collateral for variation margin, whereas the MVA is the cost of funding segregated collateral posted as initial margin.

We emphasize that we ignore the DVA and we only deal with nonnegative XVA numbers, in accordance with a shareholder-centric perspective where shareholders need be at least indifferent to a deal at a certain price for the deal to occur at that price. In case a DVA is needed (e.g. for regulatory accounting purposes), it can be computed much like the CVA. Regarding the funding issue, we consider an asymmetric, nonnegative FVA, which is part of an FVA/FDA (where FDA sits for funding debt adjustment) accounting framework, as opposed to an FCA/FBA (funding cost adjustment/funding benefit adjustment) accounting framework where it is (unduly) assumed that the bank earns its credit spread when it invests excess cash generated from trading (see [7]).

Example 3.2.1 *We consider a bank engaged into bilateral trading with a single client, with final maturity of the portfolio T. Let R_c denote the recovery rate of the client in case it defaults at time τ_c. Let* PIM *and* RIM *denote the initial margins posted and received by the bank on its client portfolio. Then, assuming for simplicity an instantaneous liquidation of the bank portfolio in case the client defaults, we have, for $0 \leq t \leq T$ (counting the* VM *positively when received by the bank) :*

$$CVA_t = \mathbb{E}_t \mathbb{1}_{\{t<\tau_c\leq T\}} \beta_t^{-1}\beta_{\tau_c}(1-R_c)\big(\mathrm{MtM}_{\tau_c} + D_{\tau_c} - D_{\tau_c-} - \mathrm{VM}_{\tau_c} - \mathrm{RIM}_{\tau_c}\big)^+, \tag{3.2.2}$$

$$FVA_t = \mathbb{E}_t \int_t^{\tau_c\wedge T} \beta_t^{-1}\beta_s\lambda_s\Big(\mathrm{MtM}_s - \mathrm{VM}_s - \mathrm{CVA}_s - FVA_s - MVA_s\Big)^+ ds, \tag{3.2.3}$$

$$\mathrm{MVA}_t = \mathbb{E}_t \int_t^{\tau_c\wedge T} \beta_t^{-1}\beta_s\bar{\lambda}_s\mathrm{PIM}_s ds. \tag{3.2.4}$$

See [11] for the extension of these equations to the case of a bank engaged into bilateral trade portfolios with several counterparties. ∎

Note that the jump process $(D_{\tau_c} - D_{\tau_c-})$ of the contractually promised cash flows contributes to the CVA exposure of the bank

$$Q = \big(\mathrm{MtM} + D - D_- - \mathrm{VM} - \mathrm{RIM}\big) \tag{3.2.5}$$

that appears in (3.2.2), because $(D_{\tau_c} - D_{\tau_c-})$ fails to be paid by the client if it defaults. In most cases, however (with the notable exception of credit derivatives exposed in Sect. 4.5), this is immaterial because $D_{\tau_c} = D_{\tau_c-}$.

Remark 3.2.3 *In the special case of a single counterparty, we also have the following equation for the $CA = CVA + FVA + MVA$ process :*

$$\begin{aligned} CA_t = \mathbb{E}_t\Big[&\mathbb{1}_{\{t<\tau_c\leq T\}}\beta_t^{-1}\beta_{\tau_c}(1-R_c)\big(\mathrm{MtM}_{\tau_c} + D_{\tau_c} - D_{\tau_c-} - \mathrm{VM}_{\tau_c} - \mathrm{RIM}_{\tau_c}\big)^+ \\ &+ \int_t^{\tau_c\wedge T} \beta_t^{-1}\beta_s\big(\lambda_s(\mathrm{MtM}_s - \mathrm{VM}_s - \mathrm{CA}_s)^+ + \bar{\lambda}_s\mathrm{PIM}_s\big)ds\Big],\ t \leq t \leq T. \end{aligned} \tag{3.2.6}$$

From the nested Monte Carlo perspective developed in Sect. 3.3, compared with (3.2.2)*, this equation "spares" the CVA and MVA Monte Carlo layer in the computation of the FVA.* ∎

At the valuation time 0, assuming deterministic interest rates, the CVA (3.2.2) can be rewritten as

$$\mathrm{CVA}_0 = (1 - R_c) \int_0^T \beta_t \mathrm{EPE}(t) \mathbb{P}(\tau_c \in dt), \tag{3.2.7}$$

where the expected positive exposure (EPE) is defined as

$$\mathrm{EPE}(t) = \mathbb{E}(Q_s^+ | s = \tau_c)_{|\tau_c = t}.$$

As explained in Sect. 4.2, this identity is popular with practitioners as it decouples the credit and the market sides of the problem. But it is specific to the valuation time 0 and is only practical when the market and credit sides of the problem are independent, so that $\mathrm{EPE}(t) = \mathbb{E}(Q_t^+)$. It can hardly be extended rigorously to wrong-way risk (cf. [49]). A similar approach is commonly applied to FVA (and DVA) computations, with analogous pitfalls.

3.2.5 Economic Capital and Capital Valuation Adjustment

On top of no arbitrage in the sense of risk-neutral CA valuation, bank shareholders need be remunerated at some hurdle rate h for their capital at risk.

The economic capital (EC) of the bank is dynamically modeled as the conditional expected shortfall (ES) at some quantile level a of the one-year-ahead loss of the bank, i.e., also accounting for discounting :

$$\mathrm{EC}_t = \mathbb{ES}_t^a \Big(\int_t^{t+1} \beta_t^{-1} \beta_s dL_s \Big). \tag{3.2.8}$$

As established in [12], Section 5.3, assuming a constant hurdle rate h, the amount needed by the bank to remunerate its shareholders for their capital at risk in the future is

$$\mathrm{KVA}_t = h \mathbb{E}_t \int_t^T e^{-\int_t^s (r_u + h) du} \mathrm{EC}_s ds, \ t \in [0, T]. \tag{3.2.9}$$

This formula yields the size of a risk margin account such that, if the bank gradually releases from this account to its shareholders an average amount

$$h(\mathrm{EC}_t - \mathrm{KVA}_t) dt, \tag{3.2.10}$$

then there is nothing left on the account at time T (if $T <$ the bank default time τ, whereas, if $\tau < T$, anything left on the risk margin account of the bank is instantaneously transferred to the creditors of the bank).

The "$-\mathrm{KVA}_t$" in (3.2.10) or the "$+h$" in the discount factor in (3.2.9) reflect the fact that the risk margin is itself loss-absorbing and as such it is part of economic capital. Hence, shareholder capital at risk, which needs be remunerated at the hurdle rate h, only corresponds to the difference (EC − KVA).

Remark 3.2.4 *For simplicity we are skipping here the constraint that capital at risk is greater than the KVA (as the risk margin is loss-absorbing). This can be formally achieved in our model by defining capital at risk as the maximum between economic capital (in the sense of the risk measure 3.2.8) and the KVA, which results in a semilinear KVA BSDE. However, due to the tendency of economic capital to decrease with time (because of the amortization of the bank portfolio modeled on run-of basis, cf. Sect. 3.2.6), taking this maximum or not typically makes no difference in practice (cf. [12], Sections 5.4 and 7.2–7.3).*

Remark 3.2.5 *An alternative to economic capital in KVA computations is regulatory capital. Regulatory capital however is less consistent, as it looses the connection with other XVAs, whereby the input to EC and in turn KVA computations should be the loss and profit process L generated by the CVA, FVA, and MVA trading of the bank (a martingale part of the CA process in* (3.2.2)*).* ■

3.2.6 Funds Transfer Price

The total (or risk-adjusted) XVA is the sum of the risk-neutral CA and of the KVA risk premium. In the context of XVA computations, derivative portfolios are typically modeled on a run-off basis, i.e. assuming that no new trades will enter the portfolio in the future. Otherwise the bank could be led into snowball Ponzi schemes, whereby always more deals are entered for the sole purpose of funding previously entered ones. Moreover, the trade-flow of a bank, which is a price-maker, does not have a stationarity property that could allow the bank forecasting future trades.

Of course in reality a bank deals with incremental portfolios, where trades are added or removed as time goes on. In practice, incremental XVAs are computed at every new (or tentative) trade, as the differences between the portfolio XVAs with and without the new trade, the portfolio being assumed held on a run-off basis in both cases. The ensuing pricing, accounting, and dividend policy ensures the possibility for the bank to go into run-off, while staying in line with shareholder interest, from any point in time onward if wished.

3.3 Multi-layered NMC for XVA Computations

In this section we present the multi-layered dependence between the different XVA metrics and we discuss their nested Monte Carlo (NMC) implementation from a high-level perspective, referring the reader to the appendix for more technical GPU implementation and optimization developments.

3.3.1 NMC XVA Simulation Tree

The EC (3.2.8) and the IM are conditional risk measures. The KVA (3.2.9) and the MVA (3.2.4) are expectations of the latter integrated forward in time (or randomized, i.e. sampled at a random future time point). The FVA (3.2.3) is the solution of a backward stochastic differential equation (BSDE). The CVA (3.2.2) and MtM are conditional expectations.

In view of Remark 3.2.5 regarding the dependence of L with respect to the CVA, the MVA, and the FVA, a full Monte Carlo simulation of all XVAs would require a five layered NMC (six layers of Monte Carlo) : The first and most inner layer would be dedicated to the simulation of MtM, then would come the IM, over which one would simulate the CVA and the MVA that are needed before simulating the FVA. The FVA layer must be run iteratively in time for solving the corresponding BSDE on a coarse time grid. All previous quantities are used in the computation of the economic capital process involved in the most outer layer that approximates the KVA.

Figure 3.1 schematizes this interdependence between the simulation of MtM and of the different XVAs. The numbers of independent trajectories that have to be resimulated at each level are denoted by M_{kva}, M_{ec}, M_{fva}, M_{cva}, M_{im}, and M_{mtm}. For computational feasability, MtM and all XVAs should be functions of time t and of a Markovian risk factor vector process Z, which can generically be taken as a Markov chain H modulated by a jump-diffusion X, where the Markov chain component H encodes the default times τ_i of all the financial entities involved : cf. the model of Section 12.2.1 in [44] (see also Section 4.2.1 there for a review of applications).

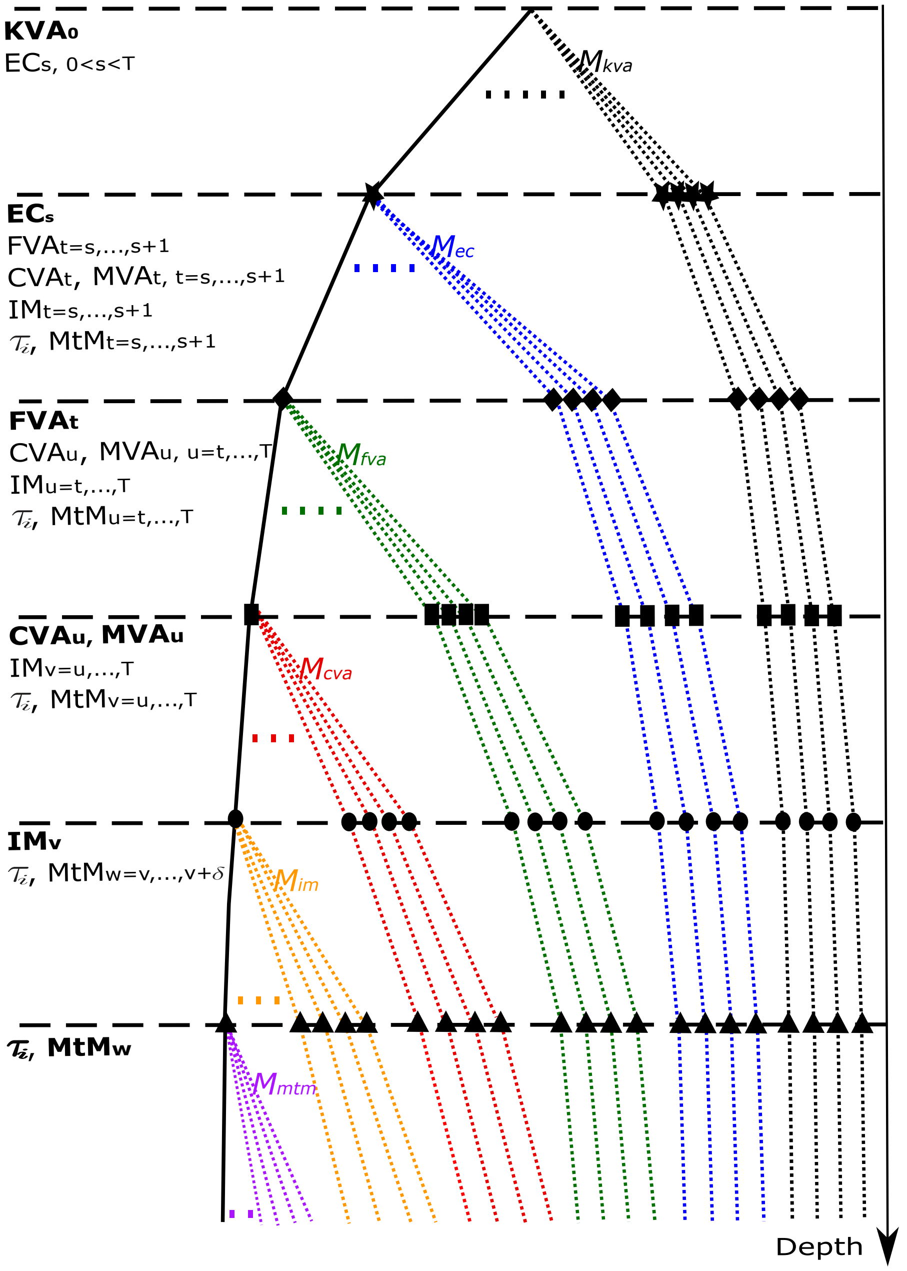


Figure 3.1 – XVA NMC simulation tree, from the most outer layer to the most inner one. The sub-tree rooted at the left-most node (on the solid path) on each line should be duplicated starting from each node on the right (on the dashed paths) on the same line. Brute force processing of the full tree would correspond to a quintuply nested Monte Carlo.

Figure 3.1 yields the overall MtM and XVAs intra-dependency structure. However : (a) if the user is only interested in some of the XVA components, then only the sub-tree corresponding to the highest XVA of interest in the figure needs be processed computationally ; (b) if one or several layers can be computed by explicit (exact or approximate) formulas instead of Monte Carlo simulation, then the corresponding layers drop out of the picture.

3.3.2 Recursive Nested vs. Iterative Proxies

The second bullet point above touches to the trade-off between iterative and recursive XVA computations, where some layers of the computation can either be recursively simulated in an NMC fashion, or iteratively pre-computed off-line and stored for further use in higher layers.

In an NMC perspective (see [68] for a seminal reference), higher layers are launched first and trigger nested simulations on-the-fly whenever required in order to compute an item from a lower layer. At the other extreme, in a purely iterative regression approach, the various layers are computed iteratively from bottom to top everywhere in time space—but with a non controllable error, as regression schemes only yield error control at the point where the regression paths are initiated. This is due to the unconditional approximations involved in regressions (unless advanced local regression basis techniques can be applied, cf. [65]). Things get even worse with wrong-way risk extensions of such an approach or with its leveraging to higher order XVAs, when several regression layers are stacked one above the other, e.g. using a one-stage simulation with the same set of trajectories for regressing globally both MtM and some XVAs. The corresponding biases can then amplify each other, resulting in quite unpredictable numerical behavior.

Beyond the XVA area, such iterative vs. recursive trade-offs are an important NMC issue. Naive parametric approximations stacked one above the other as higher order XVAs are considered can only result in uncontrollable biases. Inner simulation in nested Monte Carlo instead creates variance, which is more manageable. Another alternative is the use of deterministic pricing schemes based on probability transition matrices as in [13] and [10]. These can be very efficient and do not create any other than discretization bias, but they are restricted to certain models as discussed in Sect. 4.2.

In practice, as of today, playing with the two bullet points above, banks tend to put themselves in a position where at most one nested layer of Monte Carlo is required. For instance, 90% at least of their trading book consists of products with mark-to-market analytics, which spares the lowest layer in Figure 3.1. So far banks mostly rely on a risk weighted assets (RWA) or a regulatory capital based KVA instead of an economic capital based one (see [72]). In the context of MVA simulations, they develop various tricks such as so-called dynamic IM mapping in order to avoid resimulating for computing the IM at every future node (see [71] or [21]). They avoid the BSDE feature of an asymmetric FVA by switching to a symmetric FVA corresponding to a conditional expectation for a suitably modified discount factor, which spares the iterated regressions in FVA computations (see [11, Remark 5.1], [7], [8], [9], and [15]). Above all, most banks are using an exposure-based XVA approach, described Sect. 4.2 and in the last paragraph of Sect. 3.2.4, where it is at most the determination of the mark-to-market cube that involves (simply layered) Monte Carlo simulation(/regression).

3.3.3 XVA NMC Design Parameterization

Based on sufficient regularity as well as a non-exploding moments condition expressed by their Assumption 1, [68] show that, for balancing the outer variance and square bias components of the mean square error (MSE) in a one-layered NMC, the number of inner trajectories $M_{(1)}$ must be asymptotically of the order of the square root of the number of outer trajectories $M_{(0)}$. This is because, by a classical cancellation of the first order term in the related Taylor formula, the variance produced in the inner stage of a one-layered NMC is transformed into a bias in the outer stage. Hence, assuming that the variance of the outer random quantity is of the same order of magnitude as the conditional variances produced by the inner random quantities, an $M_{(0)} \otimes M_{(1)} = M_{(0)} \otimes \sqrt{M_{(0)}}$ NMC has the same $\mathrm{O}(M_{(0)}^{-\frac{1}{2}})$ accuracy as an $M_{(0)} \otimes M_{(1)} = M_{(0)} \otimes M_{(0)}$ NMC.

To illustrate this in an XVA setup, denoting unbiased Monte Carlo estimators of the (time 0) XVAs and of the MtM by $\widehat{\cdot}$ and referring to the dependence between XVAs or MtM in functional form, e.g. CVA(MtM), we compute :

$$\begin{aligned}\mathrm{MSE}_{cva}^2 &= \mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}) - \mathrm{CVA}(\mathrm{MtM}))^2 \\ &= \mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}) - \mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})))^2 + [\mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})) - \mathrm{CVA}(\mathrm{MtM})]^2.\end{aligned}$$

In the context of an outer CVA Monte Carlo, the first term is a variance like $\mathrm{O}(\frac{1}{M_{cva}})$. The second term is the square of a bias that can be Taylor expanded as follows (as $\mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})) = \mathbb{E}(\mathrm{CVA}(\widehat{\mathrm{MtM}})))$:

$$\begin{aligned}\mathbb{E}[\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})] - \mathrm{CVA}(\mathrm{MtM}) &= \partial_{\mathrm{MtM}}\mathrm{CVA} \times (\mathbb{E}\widehat{\mathrm{MtM}} - \mathrm{MtM}) \\ &+ \frac{1}{2}\partial^2_{\mathrm{MtM}^2}\mathrm{CVA} \times \mathbb{E}(\widehat{\mathrm{MtM}} - \mathrm{MtM})^2,\end{aligned}$$

where the first line vanishes, because $\widehat{\mathrm{MtM}}$ estimates without bias MtM, whereas $\mathbb{E}(\widehat{\mathrm{MtM}} - \mathrm{MtM})^2$ is a variance like $\mathrm{O}(\frac{1}{M_{mtm}})$. In conclusion, we obtain

$$\mathrm{MSE}_{cva}^2 = \mathrm{O}(\frac{1}{M_{cva}} + \frac{1}{M_{mtm}^2}).$$

Leveraging this idea for a two-layered NMC with homogeneous variances, nested Taylor expansions yield that an $M_{(0)} \otimes M_{(0)} \otimes M_{(0)}$ NMC has the same order of accuracy as an $M_{(0)} \otimes M_{(0)} \otimes \sqrt{M_{(0)}}$ NMC, itself as accurate as an $M_{(0)} \otimes \sqrt{M_{(0)}} \otimes \sqrt{M_{(0)}}$ NMC.

For instance, in the case of an outer FVA Monte Carlo, we have :

$$\begin{aligned}\mathrm{MSE}_{fva}^2 &= \mathbb{E}(\widehat{\mathrm{FVA}}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})) - \mathrm{FVA}(\mathrm{CVA}(\mathrm{MtM})))^2 \\ &= \mathbb{E}(\widehat{\mathrm{FVA}}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})) - \mathbb{E}(\widehat{\mathrm{FVA}}\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}))))^2 \\ &\quad + [\mathbb{E}(\widehat{\mathrm{FVA}}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}))) - \mathrm{FVA}(\mathrm{CVA}(\mathrm{MtM}))]^2.\end{aligned}$$

The term in the second line is a variance like $\mathrm{O}(\frac{1}{M_{fva}})$. The third line is the square of a bias that can be Taylor expanded as follows (as $\mathbb{E}(\widehat{\mathrm{FVA}}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}))) = \mathbb{E}(\mathrm{FVA}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}))))$:

$$\begin{aligned}&\mathbb{E}(\widehat{\mathrm{FVA}}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}))) - \mathrm{FVA}(\mathrm{CVA}(\mathrm{MtM})) = \\ &\quad \partial_{\mathrm{CVA}}\mathrm{FVA} \times (\mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})) - \mathrm{CVA}(\mathrm{MtM})) \\ &\qquad + \frac{1}{2}\partial^2_{\mathrm{CVA}^2}\mathrm{FVA} \times \mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}) - \mathrm{CVA}(\mathrm{MtM}))^2,\end{aligned}$$

where $\mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}}))-\mathrm{CVA}(\mathrm{MtM})$ was seen before to be $\mathrm{O}(\frac{1}{M_{mtm}})$, whereas $\mathbb{E}(\widehat{\mathrm{CVA}}(\widehat{\mathrm{MtM}})-\mathrm{CVA}(\mathrm{MtM}))^2$ is MSE^2_{cva}. Hence

$$\mathrm{MSE}^2_{fva} = \mathrm{O}(\frac{1}{M_{fva}} + \frac{1}{M^2_{cva}} + \frac{1}{M^2_{mtm}}).$$

As established in [105], in the general case of an $i \geq 1$-layered NMC (e.g. $i = 5$ for the overall simulation of Figure 3.1) and assuming the same variance created through the different stages, $M_{(0)} \otimes M_{(1)} \otimes \ldots \otimes M_{(i)} = M_{(0)} \otimes \sqrt{M_{(0)}} \otimes \ldots \otimes \sqrt{M_{(0)}}$ has the same $\mathrm{O}(M_{(0)}^{-\frac{1}{2}})$ accuracy as $M_{(0)} \otimes M_{(0)} \otimes \ldots \otimes M_{(0)}$.

Moreover, in practice, the variance is not homogeneous with respect to the stages : For the $\mathbb{V}\mathrm{a}\mathbb{R}$ and ES of confidence level a that correspond to the IM and EC layers, the asymptotic rate of convergence is still given by the square root of the corresponding number of simulations (as in the case of an expectation), but this may come with larger constants (proportional to $(1-\alpha)^{-1}$ in particular, see [50] and [56]). For inner layers involving time iterated conditional averaging (space regressions), such as with inner FVA or Bermudan MtM computations, one may use only $\mathrm{O}(\sqrt{M_{(0)}}/\sqrt{N_b})$ or even $\mathrm{O}(\sqrt{M_{(0)}}/N_b)$ paths (depending on the regularity of the underlying cash flows, see [1]) without compromising the overall $\mathrm{O}(M_{(0)}^{-\frac{1}{2}})$ accuracy. In view of the above, we propose an XVA NMC algorithm with the following design :

```
Input : The endowment of the bank, the credit curves of the bank
    and its clients, and, possibly, a new tentative client trade ;
Select layers of choice in a sub-tree of choice in Figure 3.1 (cf.
    Sect. 3.3.2), with corresponding tentative number of simulations
    denoted by $M_{(0)}, \ldots, M_{(i)}$, for some $1 \leq i \leq 5$ (we assume at least
    one level of nested simulation) ;
By dichotomy on $M_{(0)}$, reach a target relative error (in the sense
    of the outer confidence interval) for $M_{(0)} \otimes M_{(1)} \ldots \otimes M_{(i)}$ NMCs
    with $M_{(1)} = \ldots = M_{(i)} = \sqrt{M_{(0)}}$ ;
For each $j$ decreasing from $i$ to 1, reach by dichotomy on $M_{(j)}$ a
    target bias (in the sense of the impact on the outer confidence
    interval) for $M_{(0)} \otimes M_{(1)} \otimes \ldots \otimes M_{(j)} \otimes \ldots \otimes M_{(i)}$ NMCs ;
Return(The time 0 XVAs of interest pertaining to the bank portfolio and,
    if relevant, the incremental XVAs pertaining to a tentative trade)
```

Algorithm 5: XVA NMC algorithm.

Example 3.3.1 *Considering the overall 5-layered NMC of Figure 3.1, in order to ensure a 5% relative error (in the sense of the corresponding confidence interval) at a 95% confidence level, which we can take as a benchmark order of accuracy for XVA computations in banks, the above approach may lead to M_{mtm}, M_{im}, and M_{cva} between 10^2 and 10^3. As the FVA is obtained from the resolution of a BSDE that involves multiple (conditional) averaging, M_{fva} can be even smaller than 10^2 without compromising the accuracy. Due to the approximation of the conditional expected shortfall risk measure involved in economic capital computations, for M_{kva} of the order of 10^3, M_{ec} has to be bigger than 10^3 but usually can be smaller than 10^4.* ∎

3.3.4 Coarse and Fine Parallelization Strategies

Because of the wide exposure of a bank to a great number of counterparties, it would be very complicated and inefficient to divide the overall XVA computation into small pieces associated to local deals. In other words, it is unsuitable to distribute XVA computations : in the interest of data locality it is more appropriate to process the overall task in parallel. In fact, unless special cases (with a great deal of analytical tractability) are considered or crude and ad-hoc approximations without control error are used, the XVA NMC algorithm 5 can only be run on high performance concurrent computing architectures. Currently the most powerful ones involve GPUs (see Figure 3.4 in Section D.2). As of today, a state of the art GPU comprises roughly 4000 streaming processors operating at 2GHz each, versus 20 physical cores operating at 4GHz on a state of the art CPU (central processing unit). Hence a GPU implementation means a potential $\sim$ 100 speedup with respect to a CPU implementation.

However, execution time is an output not only of the number and nature of computations (boiling down to bitsize operations, e.g. additions), but also of the number and nature of memory accesses (depending on the technology of the related physical storage capacities as well as on the distances that separate them from the computing units on the motherboard). In practice, the main bottleneck with GPU programming is memory and, more precisely, memory bandwidth (i.e. volume of data retrievable per second from the memory).

For XVA applications this advocates the use of supercomputers with fewer and fewer but bigger and bigger computing nodes (i.e., basically, motherboards with 1 or 2 CPUs and 1 to 8 GPUs, see e.g. www.olcf.ornl.gov/summit/), equipped with a very large memory and several GPUs. However, thanks to the NVLink technology that improves the communications between GPUs (cf. [95]), scaling NMC from one GPU to various GPUs on a given node has become straightforward and very efficient. Moreover, NMC can be very well parallelized on different nodes with respect to outer trajectories, using the message passing interface (MPI) as explained in [3], [4], who compare the performances of optimized CPU and GPU implementations. Hence, nowadays, a single GPU implementation of NMC XVA computations is easily scalable to a supercomputer.

Regarding the parallelization strategy, from a GPU locality programming principle (see [94], [93]) : (a) if only one GPU is available, then the paths should be allocated between its streaming processors from the most inner nested layer of simulation to the most outer one, in a fine grain stratification approach ; (b) if several GPUs on one single node are available, then one should allocate between the different GPUs the most outer paths of the simulation, using a fine grain stratification approach on each of them for the allocation of the computational task between its streaming processors ; (c) if even several computing nodes (each equipped with several GPUs) are available, then one should allocate between them the most outer paths of the simulation and, for each of them, allocate between the corresponding GPUs the intermediary levels of the simulation, using on each of them a fine grain stratification approach for allocating more inner computations between their respective streaming processors.

Remark 3.3.1 *The first supporters of such a global procedure were [10], in the context, at that time, of CVA computations. With the advent of second generation XVAs in particular (MVA and KVA involving not only conditional expectations, but* *also conditional risk measures), data locality is even more stringent, but also much more accessible than before thanks to the large high bandwidth memories (HBM) available on GPUs and to the non-volatile memory architecture recently proposed by Intel. This favors the use of large memory computing nodes with a huge GPU computing throughput. As compared with [10] where backwardation pricing stages are treated by matrix exponentiation (in suitable models), the full NMC strategy of this paper is more complex, but it also more scalable to the above technology leaps.*

3.4 Case Studies

In this section we illustrate the above by (mostly simply) nested Monte Carlo XVA computations in various markets and setups. The corresponding GPU programming optimization techniques are detailed in the Appendix. All the models, data, and values of the numerical parameters that are used in our experiments are explicitly provided in the text or in the quoted references. Hence all these experiments are fully reproducible.

All our simulations are run on a laptop that has an Intel i7-7700HQ CPU and a single GeForce GTX 1060 GPU programmed with the CUDA/C application programming interface (API). Applying an XVA NMC computational approach to a real-life banking portfolio would require to leverage the calculations on a supercomputer with several nodes, each equipped with several CPUs and GPUs, following the strategy exposed in Sect. 3.3.4. We refer the reader to [11, Section 5] for numerical results on a real banking portfolio.

3.4.1 Common Shock Model of Default Times

Under our approach in this paper, we favor a granular simulation of all defaults other than the default of the bank (which is treated by reduction of filtration as explained in Remark 3.2.1), as opposed to working with default intensities simply.

Regarding the default times τ_i, $i = 1, \dots, n$, of the financial entities (other than the bank) involved, we use the "common shock" or dynamic Marshall-Olkin copula (DMO) model of [34], Chapt. 8–10 and [49] (see also [52], [55]). In this model defaults can happen simultaneously with positive probabilities.

First, we define shocks as pre-specified subsets of the credit names, i.e. the singletons $\{1\}, \{2\}, \dots, \{n\}$, for idiosyncratic defaults, and a small number of groups representing names susceptible to default simultaneously. For instance, a shock $\{1, 2, 4, 5\}$ represents the event that all the (non-defaulted entities among the) names 1, 2, 4, and 5 default at that time.

Given the family $\mathcal{Y}$ of shocks, the times τ_Y of the shocks $Y \in \mathcal{Y}$ are modeled as independent Cox times with intensity processes γ^Y (posibly dependent on the factor process X). For each credit name $i = 1, \dots, n$, we then set

$$\tau_i = \min_{\{Y \in \mathcal{Y};\, i \in Y\}} \tau_Y, \tag{3.4.1}$$

i.e. the default time of the name i is the first time of a shock Y that contains i. We write $J^i = \mathbb{1}_{[0,\tau_i)}$.

As demonstrated numerically Section 8.4 in [34], a few common shocks (beyond the idiosyncratic ones) are typically enough to ensure a good calibration of the model to market data regarding the credit risk of the financial entities and their default dependence (or to expert views about these).

The default model is then made dynamic, as required for XVA computations, by the introduction of the filtration of the indicator processes of the τ_i.

As detailed in Section A, we use a fine time discretization $\{t_0 = 0, \ldots, t_{N_f} = T\}$ for the forwardation of the market risk factors and a coarse discretization $\{s_0 = 0, \ldots, s_{N_b} = T\}$ embedded in the latter for prices backwardation. In practice banks use adaptative time step, e.g. $\mathrm{h}_{k+1} = \mathrm{s}_{k+1} - \mathrm{s}_k$, reflecting the density of the underlying cash flows. In our case studies we just use uniform grids, fixing the number of time steps N_f used for a fine discretization of the factor process X (and then in turn of the Cox times used for generating the τ_Y) as a multiple of some coarse discretization parameter N_b. We simulate $M_{(0)}$ outer stage trajectories from $t_0 = 0$ to $t_{N_f} = T$. From each simulated value $\widehat{X}_k$, on the coarse time grid $k \in \{1, \ldots, N_b\}$, we simulate on the fine grid $M_{(1)}$ inner trajectories starting at $s_k = t_{kN_f/N_b}$ and ending at $t_{N_f} = T$, the final maturity of the portfolio. We denote by $(\widehat{Y}_k)$ any discretization on the coarse time grid (s_k) of a continuous time process (Y_t), by $(\widehat{Y}^j_k)$ independent simulations of the latter, and by $\widehat{\mathbb{E}}$ the empirical mean.

3.4.2 CVA on Early Exercise Derivatives

First we deal with CVA and the embedded MtM computations regarding a Bermudan option on a trivariate Black–Scholes process X.

We consider a Bermudan derivative, with mark-to-market $\mathrm{MtM}_t = P(t, X_t)$ for some pricing function $P = P(t, x)$ such that, at each pricing / exercise time s_k,

$$P(s_k, X_{s_k}) = \sup_{\theta \in \mathcal{T}^k} \mathbb{E}\left(\beta_{s_k}^{-1} \beta_\theta \phi(X_\theta) | X_{s_k}\right), \tag{3.4.2}$$

where ϕ is the payoff function and $\mathcal{T}^k$ is the set of stopping times valued in the coarse time subgrid $\{s_k, \ldots, s_{N_b}\}$.

For $k = 0, \ldots, N_b - 1$, let

$$A_k = \left\{\beta_{s_k} \phi(X_{s_k}) > \mathbb{E}\left(\beta_{s_{k+1}} P(s_{k+1}, X_{s_{k+1}}) \middle| X_{s_k}\right)\right\}.$$

As detailed in [42], we have $P(0, x) = \mathbb{E}\left(\beta_{\tau_0^*} \phi(X_{\tau_0^*}) | X_0 = x\right)$, where τ_0^* is defined through the iteration $\tau_{N_b}^* = T$ and, for k decreasing from $(N_b - 1)$ to 0,

$$\tau_k^* = s_k \mathbb{1}_{A_k} + \tau_{k+1}^* \mathbb{1}_{A_k^c}. \tag{3.4.3}$$

The computation of $\mathrm{MtM}_0 = P(0, X_0 = x)$ can then be performed by a simulation/regression/projection á la Longstaff and Schwartz [90], consisting in approximating the conditional expectation involved in A_k by regression on a basis of monomial functions. Moreover, in an NMC setup, this can be extended to the estimation of any $\mathrm{MtM}_{s_{k+1}} = P(s_{k+1}, X_{s_{k+1}})$ in the obvious way, by nested simulation rooted at $(s_{k+1}, X_{s_{k+1}} = x)$.

Let $\tau_1 = \tau_c$ represent the default time of a client of the bank with default intensity equal to $\gamma^c = 100$bp and recovery $R_c = 0$ (here $n = 1$, i.e. there is only one credit name involved). The process X is a three dimensional Black & Scholes model with common volatilities $\sigma_{i=1,2,3} = 0.2$ and pairwise correlations $= 50\%$. The spot value

$X_0^{i=1,2,3} = 100 = K$, where K is the strike of a Bermudan put on the average with maturity $T = 1$. We assume no (variation or initial) margins.

By (3.2.2), the time 0 CVA of the option is approximated as

$$\mathrm{CVA}_0 \approx \sum_{k=0}^{N_b-1} \mathbb{E}\Big(\mathbb{1}_{\{\tau_c \in (s_k, s_{k+1}]\}} \beta_{s_{k+1}} (1 - R_c) \mathrm{MtM}^+_{s_{k+1}}\Big),$$

which we estimate by

$$\widehat{\mathrm{CVA}}_0 = \frac{1}{M_{cva}} \sum_{j=1}^{M_{cva}} \sum_{k=0}^{N_b-1} \mathbb{1}_{\{\tau_c^j \in (s_k, s_{k+1}]\}} \widehat{\beta}^j_{k+1} (\widehat{\mathrm{MtM}}^j_{k+1})^+, \qquad (3.4.4)$$

where the $\widehat{\mathrm{MtM}}^j_{k+1}$ are computed by nested Longstaff-Schwartz algorithms as explained above.

Using the GPU optimizations presented in Sections A, B.2, and D.1 , within less than a tenth of a second, this procedure yields the results presented in Table 3.1 for the computation of $\widehat{\mathrm{CVA}}_0$. According to Table 3.1 (note that all indications of computation times are deferred to Table 3.9), for $M_{cva} = 4096$, it is not necessary to simulate more than $M_{mtm} = 128$ inner trajectories if one targets an (absolute) error of ±5, as the corresponding estimates for $M_{mtm} = 128$ and 512 only differ by 1.

M_{mtm}	CVA_0 value	Exec. time (sec)
32	348	0.027
128	334	0.084
512	333	0.306

TABLE 3.1 – CVA_0 for an Bermudan put of payoff $(K - \frac{1}{3}(x_1 + x_2 + x_3))^+$: $M_{cva} = 4096$ and confidence interval 95% of ±5.

3.4.3 CVA and FVA on Defaultable Claims

Next we deal with CVA, FVA, and the embedded MtM computations relative to a CDS portfolio.

In this case, $\tau_1 = \tau_c$ represents the default time of a single financial counterparty (client) of the bank, with default intensity $\gamma^c = \sum_{\{Y \in \mathcal{Y};\, 1 \in Y\}} \gamma^c$ and recovery rate R_c, and $\tau_2, \ldots, \tau_n$ are the default times of reference financial entities in credit default swaps between the bank and its client.

We consider stylized CDS contracts corresponding to cash flow processes of the form, for $i \geq 2$ and $0 \leq t \leq T$:

$$D^i_t = Nom_i \times \big((1 - R_i)\mathbb{1}_{t \geq \tau_i} - S_i(t \wedge \tau_i)\big), \qquad (3.4.5)$$

where all recoveries R_i are set to 40% and all nominals Nom_i are set to 100.

A portfolio of 70 payer CDS contracts and 30 receiver CDS contracts, as well as all the numerical parameters for the simulations, are taken as in [34], Section 10.1.2, in the case of a client with CDS spread equal to 100bp. In particular, no variation or initial margins are assumed.

We denote by $\text{MtM}^i = \mathbb{E}_t[\int_t^T \beta_t^{-1}\beta_s dD^i_s]$ the mark-to-market of a fees-payer CDS on name i, so that the mark-to-market of the overall portfolio is

$$\text{MtM} = \left(\sum_{i\ pay} - \sum_{i\ rec}\right) \text{MtM}^i. \tag{3.4.6}$$

We also consider the pre-default pricing functions $P^i(t,x)$ such that $\text{MtM}^i_t = J^i_t P^i(t, X_t)$ (cf. Proposition 13.4.1 in [34]).

The contractual spreads S_i are such that the CDS contracts break even at time 0, i.e $\text{MtM}^i_0 = 0$. Setting $\text{LGD}_i = (1 - R_i)Nom_i$ (the loss on the bond of the reference name i in case it defaults, in the financial interpretation), we denote

$$\Delta_t = (\sum_{i\ \text{pay}} - \sum_{i\ \text{rec}})(D^i_t - D^i_{t-}) = (\sum_{i\ \text{pay}} - \sum_{i\ \text{rec}})\mathbb{1}_{\tau_i = t < T}\text{LGD}_i,$$

the jump process of the cash flows priced by MtM (as explained after (3.2.5), the process Δ belongs to the CVA exposure).

In order to realistically mimic the randomness of the CDS credit spreads, we use the affine intensity framework of [34], Section 10.1.1, for the intensities γ^Y in the common shocks model, based on a multivariate Cox-Ingersoll-Ross (CIR, component-wise) factor process $X = (X^1, X^2, X^3)$.

CVA

We have the following representation for the time 0 CVA (cf. (3.2.2)) :

$$\text{CVA}_0 \approx \sum_{k=0}^{N_b-1} \mathbb{E}\Big(\mathbb{1}_{\{\tau_c \in (s_k, s_{k+1}]\}}\beta_{s_{k+1}}(1 - R_c)\times$$
$$[(\sum_{i\ pay} - \sum_{i\ rec})\mathbb{1}_{\{\tau_i > s_{k+1}\}}P^i(s_{k+1}, X_{s_{k+1}}) + (\sum_{i\ pay} - \sum_{i\ rec})\mathbb{1}_{\{\tau_i \in (s_k, s_{k+1}]\}}\text{LGD}_i]^+\Big),$$

Denoting by $(\widehat{P}^{i,j}_k, \widehat{\tau}^j_i)$ simulated values of the $P^i(s_k, X_{s_k})$, estimated using inner trajectories, and of the τ_i, our estimator $\widehat{\text{CVA}_0}$ of (3.4.3) is given by (recall $\tau_1 = \tau_c$)

$$\widehat{\text{CVA}_0} = \frac{1}{M_{cva}} \sum_{j=1}^{M_{cva}} \sum_{k=0}^{N_b-1} \mathbb{1}_{\{\widehat{\tau}^j_1 \in (s_k, s_{k+1}]\}}\widehat{\beta}_{k+1}(1 - R_c) \times$$
$$\left[(\sum_{i\ pay} - \sum_{i\ rec})\mathbb{1}_{\{\widehat{\tau}^j_i > s_{k+1}\}}\widehat{P}^{i,j}_{k+1} + (\sum_{i\ pay} - \sum_{i\ rec})\mathbb{1}_{\{\widehat{\tau}^j_i \in (s_k, s_{k+1}]\}}\text{LGD}_i\right]^+.$$

Table 3.2 shows the results of a GPU implementation of the above benefiting from the optimizations presented in Sections A and B.2. We see that, for $M_{cva} = 1024*100$ paths, taking $M_{mtm} = 128$ is already enough : The gain in bias that results from taking M_{mtm} greater is negligible with respect to the uncertainty of the simulation (size of the confidence interval).

M_{mtm}	CVA value	CI 95%	Rel. err.	Exec. time (sec)
128	2358.92	±76.47	3.24%	5.96
256	2358.46	±76.45	3.24%	11.54
512	2367.90	±76.49	3.23%	23.18
1024	2373.83	±76.51	3.22%	46.12

TABLE 3.2 – CVA : $M_{cva} = 1024 * 100$, counterparty spread = 100bp.

CA BSDE

We now move to FVA computations.

Assuming no margins on client trades (i.e. VM = PIM = RIM = 0) and an instantaneous liquidation of defaulted names, the BSDE (3.2.6) for the CA = CVA+ FVA process reads as

$$\begin{aligned}\beta_t \mathrm{CA}_t = \mathbb{E}_t\Big[&\beta_{\tau_c}\mathbb{1}_{\{t<\tau_c<T\}}(1-R_c)(\mathrm{MtM}_{\tau_c}+\Delta_{\tau_c})^+ \\ &+\int_t^{\tau_c\wedge T}\beta_s\lambda(\mathrm{MtM}_s-\mathrm{CA}_s)^+ds\Big],\ t\in[0,T].\end{aligned} \tag{3.4.7}$$

For $t \in [0,T]$, let

$$\begin{aligned} cva_t &= (1-R_c)\sum_{Y\in\mathcal{Y};1\in Y}\gamma_t^Y\Big(\big(\sum_{i\ \mathrm{pay},i\notin Y}-\sum_{i\ \mathrm{rec},i\notin Y}\big)(\mathrm{MtM}_t^i+\mathbb{1}_{\tau_i=t<T}\mathrm{LGD}_i)\Big)^+ \\ f_t(y) &= \mathrm{cva}_t+\lambda(\mathrm{MtM}_t-y)^+-(r_t+\gamma_t^c)y.\end{aligned}$$

Remark 3.4.1 *Estimating $f_t(y)$ requires a Monte Carlo simulation for evaluating the embedded MtM_t^i. Hence, in an outer simulation context, obtaining an estimate $\widehat{f}_k(y)$ of $f_{s_k}(y)$ requires nested simulation.* ∎

Following the approach detailed in [49], the original CA BSDE (3.4.7) on $[0,\tau_c\wedge T]$ is equivalent to the reduced BSDE on $[0,T]$ given by

$$\mathrm{CA}_t = \mathbb{E}_t\int_t^T f_s(\mathrm{CA}_s)ds, t\in[0,T], \tag{3.4.8}$$

i.e. the solutions to (3.4.7) and (3.4.8) (both well-posed under mild technical conditions) coincide before the client default time $\tau_c = \tau_1$ and, in particular, at the valuation time 0.

The reduced CA BSDE (3.4.8) is approximated on the coarse time grid, with time steps denoted by h_{k+1} (a constant in our numerics), by $\mathrm{CA}_T = 0$ and, for k decreasing from (N_b-1) to 0,

$$\mathrm{CA}_{s_k} \approx \mathbb{E}_{s_k}\left(\mathrm{CA}_{s_{k+1}}+\mathrm{h}_{k+1}f_{s_{k+1}}(\mathrm{CA}_{s_{k+1}})\right), \tag{3.4.9}$$

where the conditional expectation is estimated by regression.

Remark 3.4.2 *For dealing with FVA semilinearity, alternatives to time iterated regressions in space are the marked branching diffusion approach of [75] or (modulo the regularity issue related to Lipchitz only coefficients as discussed in [66]) the expansion technique of [58], [59] : see [47] for a comparison between these approaches in a single layer, non nested Monte Carlo setup.*

As our forward factor process Z is very high-dimensional (three CIR processes X plus the default indicator process of each CDS reference credit name), by principal component analysis (PCA), we filter out the few eigenvectors associated with the numerically null eigenvalues, to make the PCA numerically well-posed, and we reduce further the dimension of the regression to accelerate it. In practice, keeping

90% to 95% of the information spectrum (total variance, i.e. sum of the eigenvalues) is found to require no more than half of the eigenvalues, and even much less when the time step is getting away from the maturity (cf. the right panel in Figure 3.9). We denote by $\widehat{\Psi}_k$, $\widehat{\Lambda}_k$, and $\widehat{\Gamma}_k$ the reduced regression basis, eigenvalues vector, and eigenvectors matrix that are used at each (coarse) regression time s_k. Let

$$\begin{aligned}\Phi_{s_{k+1}} &= \mathrm{CA}_{s_{k+1}} + \mathrm{h}_{k+1} f_{s_{k+1}}(\mathrm{CA}_{s_{k+1}}) - \mathbb{E}\big(\mathrm{CA}_{s_{k+1}} + \mathrm{h}_{k+1} f_{s_{k+1}}(\mathrm{CA}_{s_{k+1}}\big),\\ \widehat{\Phi}_{k+1} &= \widehat{\mathrm{CA}}_{k+1} + \mathrm{h}_{k+1}\widehat{f}_{k+1}(\widehat{\mathrm{CA}}_{k+1}) - \widehat{\mathbb{E}}\big(\widehat{\mathrm{CA}}_{k+1} + \mathrm{h}_{k+1}\widehat{f}_{k+1}(\widehat{\mathrm{CA}}_{k+1})\big).\end{aligned}$$

A centered approximation to (3.4.9) is given by

$$\mathbb{E}\big(\Phi_{s_{k+1}}|Z_{s_k}\big) \simeq \ \widehat{B}_k^{\mathsf{T}}\widehat{\Psi}_k,$$

where $\cdot^{\mathsf{T}}$ denotes matrix transposition and

$$\widehat{B}_k = \widehat{\Lambda}_k^{-1}\widehat{\Gamma}_k^{\mathsf{T}}\widehat{\mathbb{E}}(\widehat{\Phi}_{k+1}\widehat{Z}_k),\ \widehat{\Psi}_k = \ \widehat{\Gamma}_k^{\mathsf{T}}\widehat{Z}_k.$$

The ensuing CA estimator is defined by $\widehat{\mathrm{CA}}_{N_b} = 0$ and, for $k = N_b - 1, \dots, 2$,

$$\widehat{\mathrm{CA}}_k = \frac{1}{M_{fva}}\sum_{j=1}^{M_{fva}}[\widehat{\mathrm{CA}}_{k+1}^{j} + \mathrm{h}_{k+1}\widehat{f}_{k+1}^{j}(\widehat{\mathrm{CA}}_{k+1}^{j})] + \ \widehat{B}_k^{\mathsf{T}}\widehat{\Psi}_k,$$

followed by

$$\widehat{\mathrm{CA}}_0 = \frac{1}{M_{fva}}\sum_{j=1}^{M_{fva}}\big(\widehat{\mathrm{CA}}_1^{j} + \mathrm{h}_1\widehat{f}_1^{j}(\widehat{\mathrm{CA}}_1^{j})\big),$$

where each $\widehat{f}_{k+1}^{j}(\widehat{\mathrm{CA}}_{k+1}^{j})$ requires a nested simulation for evaluating the embedded $\mathrm{MtM}_{s_{k+1}}$ (cf. Remark 3.4.1).

Tables 3.3 and 3.4 show the results of a GPU implementation of the above using the optimizations of Sections A and D.2.

M_{mtm}	FVA value	CI 95%	Rel. err.	Exec. time (sec)
128	1861.66	±52.03	2.79%	81.21
256	1861.96	±52.03	2.79%	162.12
512	1861.83	±52.03	2.79%	324.67

Table 3.3 – FVA : $M_{fva} = 128 * 100$, $\lambda = 50$bp, keeping 90% of the information in the regressions for the conditional expectations.

M_{mtm}	FVA value	CI 95%	Rel. err.	Exec. time (sec)
128	1886.60	±53.75	2.87%	81.41
256	1886.73	±53.75	2.87%	162.52
512	1886.91	±53.75	2.87%	324.97

Table 3.4 – FVA : $M_{fva} = 128 * 100$, $\lambda = 50$bp, keeping 95% of the information in the regressions for the conditional expectations.

The third and fourth columns in Table 3.3 show that, when 90% of the spectrum is kept in the regressions for the conditional expectations, $M_{fva} = 128 * 100$ outer simulations are enough to ensure a reasonable accuracy. The second column shows that the FVA values are then already stabilized after a low number $M_{mtm} = 128$ of inner simulations.

Table 3.4 shows the analogous results when 95% of the information is kept in the regressions. As the confidence interval in the previous 90% case is far greater than the impact on the FVA estimate of switching from 90% to 95%, we conclude that keeping 90% of the spectrum is sufficient.

3.4.4 KVA on Swaps

Finally we consider the time 0 KVA and the embedded EC computations in the context of the sizing and allocation of the default fund of a CCP (see the last paragraph in Sect. 3.2.1). All the lower layers in Figure 3.1 are analytical in the considered (Black-Scholes) model, so that the computation can be simply nested. However we also show doubly nested Monte Carlo simulations, for illustrative purposes. The GPU implementation involves the optimizations presented in Sections A, B.1, and C.

We consider a CCP with $(n+1)$ clearing members, labeled by $i = 0, 1, 2, \dots, n$. We denote by (a) T : an upper bound on the maturity of all claims in the CCP portfolio, also accounting for a constant time $\delta > 0$ of liquidating the position between the bank and any of its counterparty in case of default ; (b) D^i_t : the cumulative contractual cash flow process of the CCP portfolio of the member i, cash flows being counted positively when they flow from the clearing member to the CCP ; (c) $\mathrm{MtM}^i_t = \mathbb{E}_t[\int_t^T \beta_t^{-1}\beta_s dD^i_s]$: the mark-to-market of the CCP portfolio of the member i ; (d) τ_i and $\tau_i^\delta = \tau_i + \delta$: the default and liquidation times of the member i, with nondefault indicator process $J^i = \mathbb{1}_{[0,\tau_i)}$; (e) $\Delta^i_{\tau_i^\delta} = \int_{[\tau_i, \tau_i^\delta]} \beta_t^{-1}\beta_s dD^i_s$: the cumulative contractual cash flows of the member i, accrued at the OIS rate, over the liquidation period of the clearing member i ; (f) $\mathrm{VM}^i_t, \mathrm{IM}^i_t \geq 0$: VM and IM posted by the member i at time t.

By contrast with the Cover 2 rule, which is purely market risk based, [24] study a broader risk-based specification for the sizing of the default fund, in the form of a risk measure of the one-year ahead loss-and-profit of the CCP if there was no default fund—loss-and-profit as it results from the combination of the credit risk of the clearing members and of the market risk of their portfolios. Such a specification can be used for allocating the default fund between the clearing members, after calibration of the quantile level a_{df} below to the Cover 2 regulatory size at time 0.

Specifically, we define the loss process of a CCP that would be in charge of dealing with member counterparty default losses through a CVA^{ccp} account (earning OIS) and capital at risk at the aggregated CCP level as, for $t \in (0, T]$ (starting from some arbitrary initial value, since it is only the fluctuations of L^{ccp} that matter, and denoting by $\boldsymbol{\delta}_{\tau_i^\delta}$ a Dirac measure at time τ_i^δ),

$$\begin{aligned}\beta_t dL^{ccp}_t = &\sum_i \big(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{VM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\big)^+ \boldsymbol{\delta}_{\tau_i^\delta}(dt) \\ &+ \beta_t(d\mathrm{CVA}^{ccp}_t - r_t \mathrm{CVA}^{ccp}_t)dt\end{aligned} \tag{3.4.10}$$

(and L^{ccp} constant from time T onward), where the CVA of the CCP is given as

$$\mathrm{CVA}^{ccp}_t = \mathbb{E}_t \sum_{t<\tau_i^\delta<T} \beta_t^{-1}\big(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{VM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\big)^+,\ 0 \leq t \leq T.$$

The ensuing economic capital process of the CCP is (cf. (3.2.8))

$$\mathrm{EC}_t^{ccp} = \mathbb{ES}_t^{a_{df}}\left(\int_t^{t+1} \beta_t^{-1}\beta_s dL_s^{ccp}\right), \tag{3.4.11}$$

which yields the size of an overall risk based default fund at the confidence quantile level a_{df}. In view of (3.4.10), we have in (3.4.11) :

$$\begin{aligned}\int_t^{t+1} \beta_s dL_s^{ccp} = & \sum_{t<\tau_i^\delta\leq t+1} \left(\beta_{\tau_i^\delta}(\mathrm{MtM}^i_{\tau_i^\delta} + \Delta^i_{\tau_i^\delta}) - \beta_{\tau_i}(\mathrm{VM}^i_{\tau_i} + \mathrm{IM}^i_{\tau_i})\right)^+ \\ & - (\beta_t \mathrm{CVA}_t^{ccp} - \beta_{t+1}\mathrm{CVA}_{t+1}^{ccp}).\end{aligned} \tag{3.4.12}$$

The KVA of the CCP estimates how much it would cost the CCP to remunerate all clearing members at some hurdle rate h for their capital at risk in the default fund, namely, for $t \leq T$ (cf. (3.2.9)),

$$\mathrm{KVA}_t^{ccp} = h\mathbb{E}_t\left[\int_t^T \beta_s e^{-hs}\mathrm{EC}_s^{ccp} ds\right], \tag{3.4.13}$$

which we estimate at time 0 by

$$\widehat{\mathrm{KVA}}_0^{ccp} = \frac{h}{M_{kva}}\sum_{j=1}^{M_{kva}}\sum_{k=0}^{N_b-1}\widehat{\beta}_{k+1}e^{-hs_{k+1}}\widehat{\mathrm{EC}}_{k+1}^{ccp}\mathrm{h}_{k+1}. \tag{3.4.14}$$

For our simulations we consider the CCP toy model of [24], Section 4, where $n+1=9$ members are clearing swaps on a Black underlying, with all the numerical parameters used there (in particular, $a_{df} = 99\%$).

Simply layered NMC

The time 0 KVA^{ccp} in (3.4.14) is computed in two alternative ways : (a) by a non-nested proxy approach where $\mathbb{ES}_t^{a_{df}}$ is replaced by $\mathbb{ES}_0^{a_{df}}$ in (3.4.11) and $\widehat{\mathrm{EC}}_{k+1}^{ccp}$ is replaced by the corresponding (deterministic) term structure, which we denote by $\widetilde{\mathrm{EC}}_{k+1}^{ccp}$, in (3.4.14) ; (b) by a GPU nested Monte Carlo procedure (one layered NMC) using the optimized sorting procedure of Section C for the inner $\widehat{\mathrm{EC}}_{k+1}^{ccp}$ computations.

We refer the reader to Equation (33) and the ensuing discussion in [11] for the detail of the second procedure, motivated by the fact that, for the computation of the KVA, an outer (unconditional) expectation will be applied anyway : The comparison between Tables 3.5 and 3.6–3.7 shows that the ensuing bias on the resulting time 0 KVA estimate, denoted by $\widetilde{\mathrm{KVA}}_0^{ccp}$, may be acceptable, bearing in mind the model risk intrinsic to such quantities.

Regarding the nested computation, Table 3.6 shows that the variance is acceptable already for a low number, such as $M_{kva} = 512$, of outer simulations. Table 3.7 shows that this is not specific to the relatively high number $M_{ec} = 1024 \times 100$ of inner simulations used in Table 3.6 : as we can see from Table 3.7, for $M_{kva} = 1024$, the KVA value is already stabilized for $M_{ec} = 128 \times 100$.

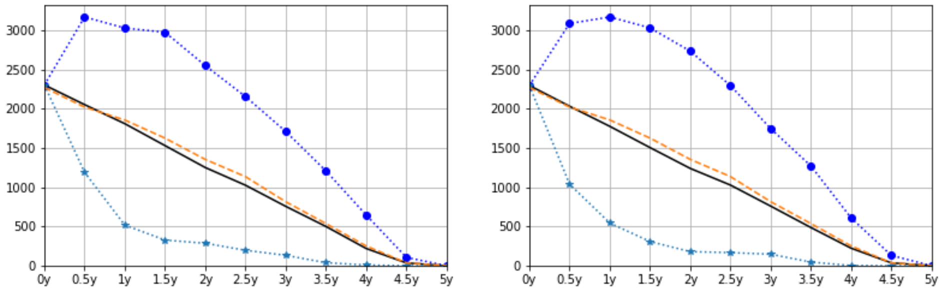


FIGURE 3.2 – Economic capital processes for $a_{df} = 99\%$. The '- -' curves correspond to a term structure approximation (cf. (3.4.11)) $\mathrm{EC}_t^{ccp} = \mathbb{ES}_t^{a_{df}}\left(\int_t^{t+1} \beta_t^{-1}\beta_s dL_s^{ccp}\right) \approx \mathbb{ES}_0^{a_{df}}\left(\int_t^{t+1} \beta_t^{-1}\beta_s dL_s^{ccp}\right)$, $0 \le t \le T$. The '$\cdots *$', '—' , and '$\cdots \bullet$' curves respectively correspond to the 1% quantile, mean, and 99% quantile functions of the random process EC^{ccp}. Left : $M_{kva} = 256$ and $M_{ec} = 32 * 100$. Right : $M_{kva} = 1024$ and $M_{ec} = 32 * 100$.

M_{kva}	$\widehat{\mathrm{KVA}}_0^{ccp}$	Exec. time (sec)
$128 * 100$	518.44	1.42
$256 * 100$	515.49	2.94
$512 * 100$	512.94	5.88
$1024 * 100$	508.96	11.74

TABLE 3.5 – KVA_0 estimated by deterministic projections of economic capital, without nested simulation.

M_{kva}	$\widehat{\mathrm{KVA}}_0^{ccp}$	CI 95%	Rel. err.	Exec. time (sec)
256	493.68	± 11.80	2.39%	0.09
512	487.53	± 8.82	1.80%	0.17
1024	489.03	± 6.07	1.24%	0.34
2048	490.33	± 4.19	0.85%	0.73
4096	491.55	± 3.01	0.61%	1.24

TABLE 3.6 – KVA_0 estimated by nested simulation : $M_{ec} = 32 * 100$.

M_{ec}	$\widehat{\mathrm{KVA}}_0^{ccp}$ value	Exec. time (sec)
$8 * 100$	479.89	0.09
$16 * 100$	485.81	0.21
$32 * 100$	489.03	0.46
$64 * 100$	490.39	1.09
$128 * 100$	491.32	3.14

TABLE 3.7 – KVA_0 estimated by nested simulation : $M_{kva} = 1024$.

Doubly layered NMC

Instead of using the explicit formulas that are available in our CCP setup, we can use second layers of NMC for estimating the CVA terms in the second line of

(3.4.12). Figure 3.3 shows the impact of the CVA terms on EC^{ccp} computed in these alternative ways when $a_{df} = 85\%$ (for $a_{df} = 99\%$ as before, the impact of the CVA fluctuations is negligible in (3.4.12)). The corresponding time 0 KVA computations are reported in Table 3.8.

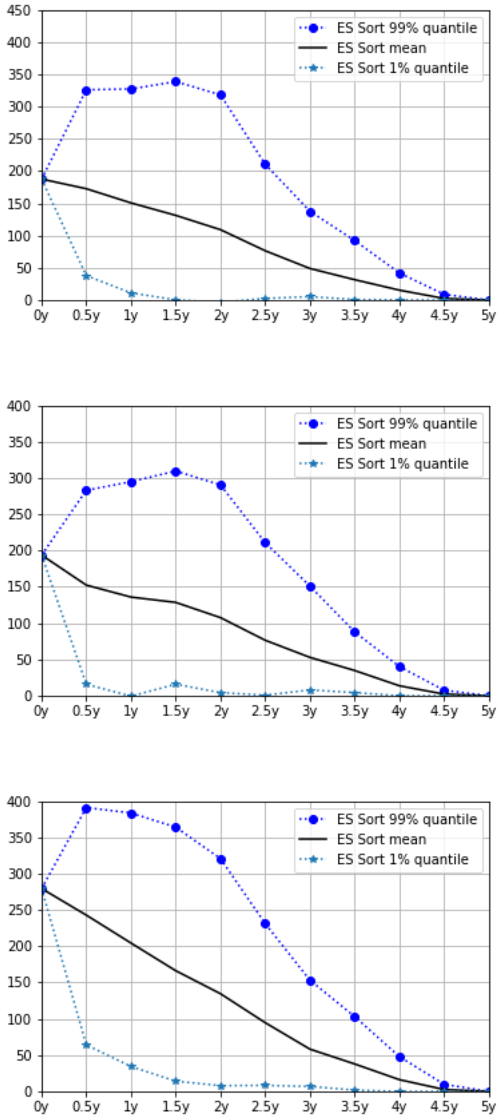


FIGURE 3.3 – For $M_{kva} = 1024$ and $M_{ec} = 32 * 100$: Mean and 1% and 99% quantile functions of the EC^{ccp} process for $a_{df} = 85\%$, computed [top] by simple NMC, using the explicit formula for the embedded CVA computations, [middle] by double NMC, using simulation with $M_{cva} = 32$ for the embedded CVA computations, [bottom] by simple NMC ignoring the CVA terms, i.e. ignoring the second line in Eq (3.4.12).

3.4.5 Speedup Table

GPU optimizations, regarding a proper use of the different available memory layers in particular, are important for effectively benefiting from the target ~ 100 speedup factor when compared with an optimized CPU implementation. Table 3.9 summarizes the speedups obtained in our case studies thanks to the various GPU

	$\widehat{\mathrm{KVA}}_0^{a,ccp}$			$\widehat{\mathrm{KVA}}_0^{b,ccp}$			$\widehat{\mathrm{KVA}}_0^{c,ccp}$		
M_{kva}	Value	CI 95%	Time(sec)	Value	CI 95%	Time(sec)	Value	CI 95%	Time(sec)
256	54.55	1.50	0.13	20.12	0.70	0.15	22.60	0.95	4.16
512	54.30	1.07	0.27	20.12	0.51	0.28	21.25	0.63	8.64
1024	54.10	0.75	0.53	20.12	0.31	0.6	20.31	0.42	16.9

Table 3.8 – KVA_0 computed by nested simulation for $a_{df} = 85\%$, $M_{kva} = 1024$, and $M_{ec} = 32 * 100$: $[\widehat{\mathrm{KVA}}_0^{a,ccp}]$ without the CVA term ; $[\widehat{\mathrm{KVA}}_0^{b,ccp}]$ with the CVA terms computed by explicit formula ; $[\widehat{\mathrm{KVA}}_0^{c,ccp}]$ with the CVA terms computed by MC with $M_{cva} = 32$.

optimizations detailed in the appendix. The rows of the table are labeled as the corresponding sections in the appendix.

First, the computation times also displayed in the table show that NMC computations are within reach provided GPU and the related optimizations are used, even for computations involving portfolios of one hundred credit names, time backwardation of nonlinearities, or conditional risk measure computations : Although the corresponding two-stage Monte Carlos would be quite heavy to implement on a CPU, the GPU implementations, suitably optimized, are quite fast. Additional speedups would readily follow from the use of several GPUs as explained in Sect. 3.3.4.

	CVA		FVA		KVA	
	Time	Speedup	Time	Speedup	Time	Speedup
A. Nested simulation	5.80	3.5	80.51	3.5	0.96	3.5
B.1. Sorting defaults					0.22	1.2
B.2. Listing defaults	0.06	1.4				
C. $\mathbb{V}\mathrm{a}\mathbb{R}$ and $\mathbb{ES}$					0.12	20
D.1. Inner regressions	0.06	2				
D.2. Outer regressions			0.05	3		
Other	0.1		0.64		0.11	

Table 3.9 – Execution times (in seconds) and GPU optimization speedups in the NMC case studies of Sect. 3.4.2 through 3.4.4 : CVA for $M_{cva} = 1024 * 100$ and $M_{mtm} = 128$; FVA for $M_{fva} = 128 * 100$ and $M_{mtm} = 128$; KVA for $M_{kva} = 128$ and $M_{ec} = 1024 * 100$.

The second take-away message is that the optimization speedups are quite significant, as expected. We emphasize that we are only considering here speedups internal to a GPU implementation more or less optimized in a way or another as detailed in the appendix, we do not report on any CPU implementation execution times (that would be much slower) or GPU versus CPU speedups (as an optimized GPU implemetation obviously dominates an optimized CPU implementation).

Last, one should not overdue the conclusions that can be drawn from such table. The observations, for instance, that defaults' listing and sorting would be less useful than the other optimizations, or that the inner regression phase is much faster than the nested simulation of paths, are model dependent and restricted to the setup and implementation of the case studies that underlie all numbers in Table 3.9. Thus, a careful reading of Section B shows that the relative importance of the defaults' listing and sorting optimizations strongly depends on the default structure of the considered model. Likewise, a numerically more robust implementation of the inner regressions on few paths (hence prone to ill-conditioning, numerical instabilities and sensitivity to roundoff errors), based on a spectral decomposition of the covariance matrices that are repeatedly generated along inner simulation/regression stages (see the end of Section D.1 and cf. also Sect. 3.4.3), could easily be 10 to 20 times slower than Cholesky that was used in the inner regression case study of Sect. 3.4.2. Etc..

3.5 Perspectives

Due to the number of market input data involved in the calibration of a real-life XVA engine, the consideration not only of the XVAs themselves, but also of all the corresponding bump sensitivities, may easily increase the XVA computational burden by a factor of ~ 1000. Hence, another interesting topic left aside in this paper for length sake is XVA Greeks and their GPU acceleration, whether performed by the maximum likelihood method (see [13]), by (hard to handle memory-wise though) adjoint differentiation (see [106]), or through various regression techniques. As initial margins, at least in a bilateral SIMM setup, involve a sensitivity $\mathbb{V}\mathrm{a}\mathbb{R}$, this is also the reason why we did not show any MVA computations in our case studies (MVA in a centrally cleared setup however can be tackled by similar tokens as the KVA of Sect. 3.4.4).

By their nature, XVA computations appeal naturally to nested Monte Carlo computations. However, whenever non- (or less-)nested Monte Carlo schemes are available without bias, they should be preferred to a nested alternative. In particular, the parameterized stochastic approximation or “quasi-regression” algorithm (borrowing the terminology from [14]) of [46], initially conceived for dealing with model uncertainty, could also be used for reconstructing a conditional risk measure such as $\mathbb{V}\mathrm{a}\mathbb{R}$ (for IM computations) or $\mathbb{ES}$ (for EC computations) as a function of the risk factors (provided their number can be kept reasonable), with global error control (cf. Sect. 3.3.1). To reduce further the complexity of the NMC structure, one might also think of multi-level techniques à la [61].

Banks can use their economic capital as variation margin. Accounting for this feature results in the following modified FVA formula, instead of (3.2.3) :

$$\mathrm{FVA}_t = \mathbb{E}_t \int_t^T \beta_t^{-1}\beta_s\lambda_s\Big(\mathrm{MtM}_s - \mathrm{VM}_s - \mathrm{CVA}_s - \mathrm{FVA}_s - \mathrm{MVA}_s - \mathrm{EC}_s(L)\Big)^+ ds,$$

where we recall from Sect. 3.2.4 that the trading loss process L is a martingale component of the CA = CVA+FVA+MVA process. The ensuing XVA equations “of the McKean type” are shown to be well-posed in [45]. To solve them numerically, [11] resort to a Picard iterative scheme, where each iteration is similar in nature to Figure 3.1 (or part of it, cf. Sect. 3.3.2), combined with a proxy approach for the $\mathrm{EC}_s(L)$ process, replaced by its unconditioned version as considered in Sect. 3.4.4. An unbiased approach to these equations would be another topic of future research.

APPENDIX. XVA NMC GPU Programming Optimization Techniques

Although XVA NMC computations seem naively suited to parallel architectures like graphic programming units (GPUs), a XVA NMC GPU implementation requires various computational complexity and memory storage optimizations, which are essential to achieve the potential ~ 100 speedup with respect to an optimized CPU implementation. These XVA NMC GPU programming optimization techniques are detailed in the remaining sections of the paper.

First, we need to provide some details on the GPU hardware/software aspects, referring the reader to [94] for a detailed documentation. Originally developed for gaming, beginning from around 2007, GPUs became extensively used for simulation. The most important specifications that make GPU architecture highly suited to NMC are (see Figure 3.4) : (a) A very large number of processing units grouped in streaming multiprocessors (SM). These units provide the sufficient computational power to process in parallel independent tasks ; (b) A large bandwidth to read/write data on the GPU random access memory (RAM) known as global memory. An important bandwidth is crucial to Monte Carlo since the performed operations are generally light when compared to the time spent to access to the memory ; (c) A cached memory space shared between (the 128) processors of the same streaming multiprocessor. This allows a very fast memory storage of arrays shared by various processors and makes possible the communication between them.

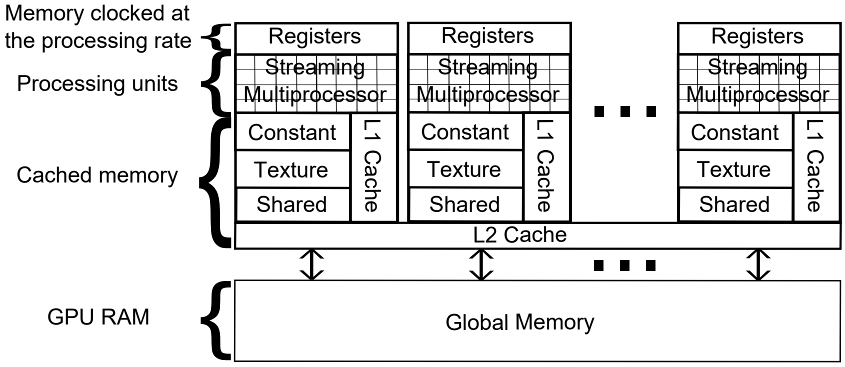


FIGURE 3.4 – Simplified presentation of the Nvidia GPU architecture.

From the computational point of view, the streaming multiprocessors (SMs) execute blocks of (less than 1024) successive tasks known as threads. Once a block is associated to an SM, each streaming processor performs the required computations of various threads from the block. Within a block, the threads are organized in warps of 32 threads executed at the same time. Threads of the same warp are synchronized automatically at the hardware level, which is together convenient (in the sense that the programmer does not need care about it) and efficient. However, because of this intrinsic synchronization among threads of the same warp, the programmer has to ensure that the number of threads in a waiting state is as small as possible. Waiting states can be caused by conditional sentences like *if* clauses.

From the storage point of view, the global memory plays the role of the GPU RAM. Because it is the largest memory space on the GPU (from 1GB to 16GB), global memory is generally used by default. However, in order to reduce the time of reading/writing values, instead of the global memory, one should **use registers and shared memory as much as possible**. As opposed to the global memory, shared memory and registers can only be accessed by their own streaming multiprocessors. Registers are the fastest and they are generally used for local variables. Although slower than registers, shared memory stores arrays that can be handled by all threads of the same block.

These are the main features that have made GPUs suitable, for more than a decade, to parallel computing and deep learning algorithms, resulting in their extensive use for artificial intelligence (see [43]). We refer the reader to [94] regarding further generic benefits of GPUs architecture in terms of concurrent execution, asynchronous data transfer between GPU and CPU, mapping the CPU memory, shuffles, tensor cores (introduced recently), etc..

There are many optimizations that must be considered in GPU programming but are not specific to NMC for XVA. The three most common ones consist in (a) ensuring that access to the RAM is as coalescing as possible, (b) reducing divergence in the code, and (c) using the constant memory (cf. Figure 3.4), very fast as read-only, for model parameters.

RAM access coalescence and divergence management are as important as in a standard parallel implementation on CPU and we refer to [94], [93] for further details. As for the constant memory requirement, it is not so important when the model is relatively simple so that the time spent in operations dominates the time spent to read the parameters. However, it becomes essential when the model involves a lot of parameters, e.g. with a local volatility model based on spline interpolation. In such cases, storing the parameters in a read-only memory considerably reduces the time spent to read the parameters and the execution time as a consequence.

In the developments that follow we mostly focus on the use of warps, registers, shared and global memories. For each optimization, we evaluate the speedup $=$ $T_{\text{simp}}/T_{\text{opti}}$, where T_{opti} is the execution time of the optimized part of the code and T_{simp} is the execution time of the same part of the code but without the considered optimization.

3.6 Optimizations Related to the Time Grids Used in Factors Forwardations and Prices Backwardations

In this section, we present straightforward but very important optimizations needed for the nested simulation of any stochastic process on GPUs. Without loss of generality, let X be the univariate diffusion (local volatility model)

$$dX_t = X_t(b(t, X_t)dt + \sigma(t, X_t)dW_t),\ X_0 = x, \tag{3.6.1}$$

where W is a Brownian motion. To compute a derivative price Y driven by X, we first have to discretize several paths of the stochastic differential equation (SDE) (3.6.1) on a fine forwardation grid $\{0 = t_0, \ldots, t_{N_f} = T\}$. Then, given a backwardation subset $\{0 = s_0, \ldots, s_{N_b}\}$ of $\{t_0, \ldots, t_{N_f}\}$, one has to approximate

$$Y_{s_k} = \mathbb{E}\left(\phi(X_{s_{N_b}})|X_{s_k}\right), \text{ for every } 0 \leq k < N_b, \tag{3.6.2}$$

for a number of payoff functions ϕ.

In practice, the fine discretization $\{t_0, \ldots, t_{N_f}\}$ used for the forwardation of the underlying risk factors can be of the order of 100 time points per year in order to ensure an acceptable time discretization error to the underlying SDEs, whereas, in order to spare computational time and avoid an exploding accumulation of regression errors in case of FVA computations (or MtM computations for American claims), the coarse discretization $\{s_0, \dots, s_{N_b}\}$ used for prices backwardation can be of the order of 10 time points per year. Hence the coarse discretization $\{s_0, \dots, s_{N_b}\}$ is $N_f/N_b \sim 10$ times smaller than $\{t_0, \dots, t_{N_f}\}$.

Be it for computing the European derivative price (3.6.2), its American version, or even risk measures conditional on the value of $X_{s_k}, 0 \leq k < N_b$, one has first to store various realizations $\{X_{s_k}\}_{0 \leq k < N_b}$ on the GPU RAM (or even on the CPU RAM if the GPU RAM is not sufficient). All the discretized intermediate realizations of $\{X_t\}_{t \in \{t_0, \dots, t_{N_f}\} \backslash \{s_0, \dots, s_{N_b}\}}$ should only be stored temporarily in the shared memory. We use shared memory that is faster than global memory. Morever, it is generally impossible to use only registers for this purpose, since one needs to store arrays of values in the case of multidimensional models. Registers, however, contain the state vector of the random number generator (RNG) used for simulating these temporary realizations.

More precisely, between each successive values $\{X_t\}_{t=s_k=t_{k'}, s_{k+1}=t_{k''}}$ stored in the global memory, one simulates temporarily $\{X_t\}_{t \in \{t_{k'+1}, \dots, t_{k''-1}\}}$ stored in shared memory, which supposes the generation of various uniformly distributed random numbers stored in a register as the RNG state vector. Even though the RNG state is given by a vector, its size is fixed and thus does not have to be stored in an array. The RNG that we use is a CMRG which is a combination, presented in [85], of two multiple recursive generators (MRGs). The first adaptation of CMRG to GPUs was proposed in [1,3].

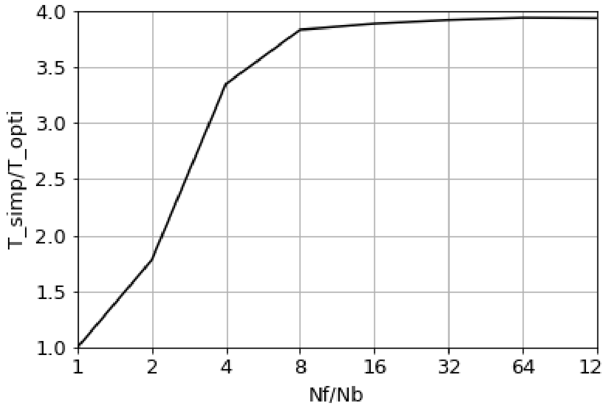


FIGURE 3.5 – The speedup, benefiting to all case studies (here in the context of the CVA and FVA case studies of Sect. 3.4.3), from using registers and shared memory during the nested simulation of the factor process X as a function of N_f/N_b increasing from 1 to 128, for N_f fixed to 128 (i.e. for N_b decreasing from 128 to 1).

For $N_f = 128$, Figure 3.5 shows the speedup obtained from this strategy when $N_f/N_b = 1, \dots, 128$ (i.e. $N_b = 128, \dots, 1$). In the right part of Figure 3.5, the speedup tanks, since the application becomes less memory bound than compute bound. In the standard situation where $N_f/N_b \sim 10$, the speedup is generally greater than 3.5.

3.7 Optimizations Related to Default Times

Be it introduced by the XVA itself or by defaultable claims involved in the underlying market exposure, indicator functions on default events $\{\tau_i \in A\}$ increase the complexity of an efficient parallel implementation. For the first situation, presented in Section B.1, we aim at reducing the GPU memory occupation for limiting memory access and thus accelerating the execution. The optimization studied in Section B.2 shows how to reduce thread divergence in order to take advantage of the maximum computing power of GPUs. The situation faced in both cases is

$$\sum_{i \in \mathcal{S}} \mathbb{1}_{\tau_i \in A} Y^i, \tag{3.7.1}$$

where $\mathcal{S}$ is a set of obligors and τ_i can be simulated before Y^i. We keep using registers and shared memory as often as possible for both τ_i and Y^i simulations.

3.7.1 Sorting Defaults for Multiple Counterparties

Here $\mathcal{S}$ is a set of bank clients with associated counterparty exposures $\phi_i = \phi_i(t,x)$, the set A is a bounded time interval $(s_k, s_{k'})$ (with $k' > k$), and $Y^i = \phi_i(\tau_i, X_{\tau_i})$. Hence (3.7.1) can be rewritten as

$$\sum_{\text{bank clients}} \mathbb{1}_{s_k < \tau_i \leq s_{k'}} \ \phi_i(\tau_i, X_{\tau_i}). \tag{3.7.2}$$

If we had to simulate the process X on the fine discretization grid and then read the memory using τ_i, the quantity of memory needed would be quite significant, especially with NMC in view. Consequently, it is essential to only store in the global memory the values taken by the X_{τ_i} (such that $s_k < \tau_i \leq s_{k'}$), which can only be done after sorting the τ_i. This method spares the use of the global memory and reduces the number of writings to it. Due to the reduced number of writings, we also speed up the execution time, in particular as long as few defaults occurred, i.e., given the (low) intensity models that we use for the τ_i, during the first time steps of the algorithm : see the left panel in Figure 3.6.

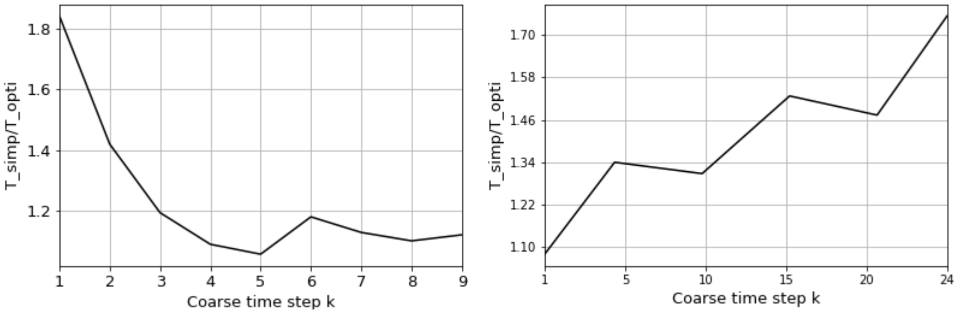


Figure 3.6 – The speedup obtained from using : [left] Sorting defaults and restricting some of the computations to already defaulted names whenever applicable, e.g. in the context of the KVA case study of Sect. 3.4.4 ; [right] Listing defaults and restricting some of the computations to the surviving names whenever applicable, e.g. in the context of the CVA case studies of Sect. 3.4.2 and Sect. 3.4.3.

3.7.2 Listing defaults for defaultable claims

Now $\mathcal{S}$ is a set of reference obligors, $A = (s_k, \infty)$, and $Y^i = P_i(s_k, X_{s_k})$ for some coarse time s_k, where $P_i = P_i(t, x)$ is a suitable (pre-default) pricing function. The expression (3.7.1) reads as

$$\sum_{i \in \text{set of obigors}} \mathbb{1}_{\tau_i > s_k} \, P_i(s_k, X_{s_k}), \tag{3.7.3}$$

where $P_i(s_k, X_{s_k})$ is assumed more difficult to simulate than τ_i, e.g. because it requires another Monte Carlo simulation.

If we associate a thread to each realization of $\{\tau_i\}_{i \in \text{set of obligors}}$, then it is possible to have some threads returning for obligor $i = 1$ (say) $\mathbb{1}_{\tau_1 > s_k} = 1$ and other threads from the same warp returning $\mathbb{1}_{\tau_1 > s_k} = 0$. This situation creates thread divergence within the same warp, forcing threads/processors with $\mathbb{1}_{\tau_1 > s_k} = 0$ to wait for the others.

To overcome this divergence, we list for each thread all the surviving obligors in order to make all threads computing together most of the time (there is always a small divergence due to the number of surviving obligors which is different between threads). Thus, the simulation of (3.7.3) is reduced to the simulation of

$$\sum_{i \in \text{list of surviving obigors}} P_i(s_k, X_{s_k}). \tag{3.7.4}$$

The corresponding speedup is demonstrated by the right panel in Figure 3.6. We see that the benefit of this strategy increases with the time index. This can be explained by the accumulated number of defaults that also increases with time, resulting in more divergence unless the above optimization is in place.

3.8 Optimized Sorting for $\mathbb{V}\mathrm{a}\mathbb{R}$ and $\mathbb{ES}$ Computations

[112] present a survey of the main GPU based sorting algorithms : radix sort, merge sort, sample sort and quick sort. However, sorting for computing risk measures such as $\mathbb{V}\mathrm{a}\mathbb{R}$ or $\mathbb{ES}$ is reduced to finding the $|\mathfrak{Q}|$ largest values from a list of $|\mathfrak{L}|$ values with $|\mathfrak{L}| \gg |\mathfrak{Q}|$, where $|\cdot|$ denotes the cardinality. This problem is less complex than sorting the whole list $\mathfrak{L}$ and it shares some similarities with the problem of finding the k nearest neighbours discussed, for instance, in [111]. The latter paper is based on merge sort, with complexity $\mathrm{O}(|\mathfrak{L}| \log |\mathfrak{L}|)$, the parallelization of which is the most suited to GPUs (see [73]).

Here, we present an original method that allows to find the $|\widetilde{\mathfrak{L}}|$ largest values from a list of $|\mathfrak{L}|$ values in very few computations, of complexity $\mathrm{O}(|\mathfrak{L}|)$. The sublist $\widetilde{\mathfrak{L}}$ satisfies $|\mathfrak{Q}| \leq |\widetilde{\mathfrak{L}}| \leq 2|\mathfrak{Q}| \leq |\mathfrak{L}|$. It is obtained from $\mathfrak{L}$ thanks to successive truncations that discard all the values smaller or equal to some threshold. One can then apply merge sort on $\widetilde{\mathfrak{L}}$ to get $\mathfrak{Q}$ with an overall complexity $\mathrm{O}(|\mathfrak{L}| + |\widetilde{\mathfrak{L}}| \log |\widetilde{\mathfrak{L}}|)$.

Denoting the median, mean, and standard deviation by μ, m, and dev, the main idea for obtaining at low cost $\widetilde{\mathfrak{L}}$ is based on Mallows inequality $|\mu - m| \leq \text{dev}$, which allows using the (computationally cheap) m and dev instead of the (sort intensive) μ in order to pre-sort $\mathfrak{L}$: Basically (see Algorithm 6), starting from $\mathfrak{L}$, the list is truncated successively (4 or 5 times are typically enough in practice) by keeping the values superior or equal to $(m + \mathrm{dev})$, m, or $(m - \mathrm{dev})$. Because we only need to compute m and dev, which have a linear complexity, the complexity of pre-sorting $\mathfrak{L}$ to obtain $\widetilde{\mathfrak{L}}$ such that $|\mathfrak{Q}| \leq |\widetilde{\mathfrak{L}}| \leq 2|\mathfrak{Q}|$ is $\mathrm{O}(|\mathfrak{L}|)$.

```
Input : A large array $\mathfrak{L}$ of unordered values, the size $|\mathfrak{Q}|$ of $\mathfrak{Q}$;
$\widetilde{\mathfrak{L}} = \mathfrak{L}$;
while $|\widetilde{\mathfrak{L}}| > 2|\mathfrak{Q}|$ do
    Compute the mean $m$ and the standard deviation dev of $\widetilde{\mathfrak{L}}$;
    if the number of values in $\widetilde{\mathfrak{L}}$ that are bigger than $m + \mathrm{dev}$ is $\geq |\mathfrak{Q}|$ then
        New $\widetilde{\mathfrak{L}} =$ values bigger than $m + \mathrm{dev}$ of the previous $\widetilde{\mathfrak{L}}$
    else if the number of values in $\widetilde{\mathfrak{L}}$ bigger than $m$ is $\geq |\mathfrak{Q}|$ then
        New $\widetilde{\mathfrak{L}} =$ values bigger than $m$ of the previous $\widetilde{\mathfrak{L}}$
    else if the number of values in $\widetilde{\mathfrak{L}}$ bigger than $m - \mathrm{dev}$ is $\geq |\mathfrak{Q}|$ then
        New $\widetilde{\mathfrak{L}} =$ values bigger than $m - \mathrm{dev}$ of the previous $\widetilde{\mathfrak{L}}$
Use merge sort on $\widetilde{\mathfrak{L}}$ to get $\mathfrak{Q}$
Return(The array $\mathfrak{Q}$.)
```

Algorithm 6: Optimized sorting algorithm for $\mathbb{V}\mathrm{a}\mathbb{R}$ and $\mathbb{ES}$ computations.

When parallelized on GPUs, the operations presented in Algorithm 6 need be synchronized between threads. As we are performing NMC with a large number of different lists $\mathfrak{L}$ on which we need to compute some risk measure, we dedicate one warp of 32 threads to each list $\mathfrak{L}$. This solution is very convenient as threads of the same warp are automatically synchronized at the hardware level. Moreover, this approach is not memory intensive, hence it scales well and is not limited by the size of the shared memory as in [111].

Figure 3.7 shows the speedup obtained by this approach when compared to a straightforward implementation of complexity $\mathrm{O}(|\mathfrak{Q}| \times |\mathfrak{L}|)$ for finding the $|\mathfrak{Q}|$ largest values among $|\mathfrak{L}|$ values.

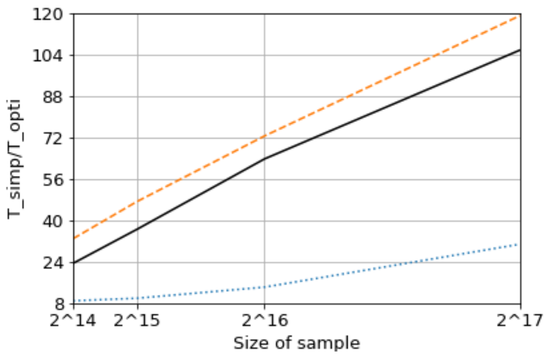


FIGURE 3.7 – The speedup obtained in the KVA case study of Sect. 3.4.4 by the application of the optimized sorting and value at risk / expected shortfall computation strategy based on a preliminary computation of the sublist of the '- -' 1% largest values, '—' 5% largest values, or '···' 10% largest values of a full sample list $\mathcal{L}$.

3.9 Regressions

Simulation/regression methods are widely used in mathematical finance. They occur in the context of Longstaff-Schwartz type dynamic programming (DP) algorithms that can be used for time 0 valuation of an early exercise derivative (see e.g. [1], [42], [90]). Such algorithms can also be used for obtaining a proxy of the totality of a European pricing function (cf. [48]), but we emphasize that, as explained in Sect. 3.3.1, this comes without global error control.

Simulation/regression schemes are also commonly used for the resolution of nonlinear BSDEs (see [65]), such as the one that appears in asymmetric FVA computations. Other examples in the XVA context (subject to the same "local versus global error control" reservation as above though) include regression schemes that may be used for capturing conditional risk measures in the context of MVA or KVA computations (see e.g. [21, 71, 91]).

For a good overview on the kind of convergence associated to regression methods, we refer the reader to [60]. As explained in [92], when the underlying risk factor process X has a density bounded away from zero (which can always be achieved numerically by truncation of X), the asymptotic number of trajectories has to be of the order of the cube of the cardinality of the regression basis, which we refer to as the cubic rule below.

3.9.1 Regressions on Inner Trajectories

When the XVA implementation requires a regression on inner NMC trajectories, this typically means relatively few paths as, on one hand, the computational cost duplicated on the inner nodes would be too large otherwise and, on the other hand, a non-negligible inner regression error is acceptable since it will be mitigated by outer averaging anyway.

The main ingredient relies on the resolution of a large (because of "inner" and of (even coarse) time stepping) number of small (because few regressors are typically used in finance, especially here due to the cubic rule in regard of the (relatively small) number of (inner) trajectories) random symmetric linear systems. For instance (cf. Figure 3.8), in a simply nested setup, the implementation of a Longstaff-Schwartz algorithm on the inner trajectories requires $(N_b - k - 1)$ regressions at each coarse time $s_k, k = 1, \ldots, N_b - 1$, and for each outer trajectory $l \in \{1, \ldots, M_{(0)}\}$.

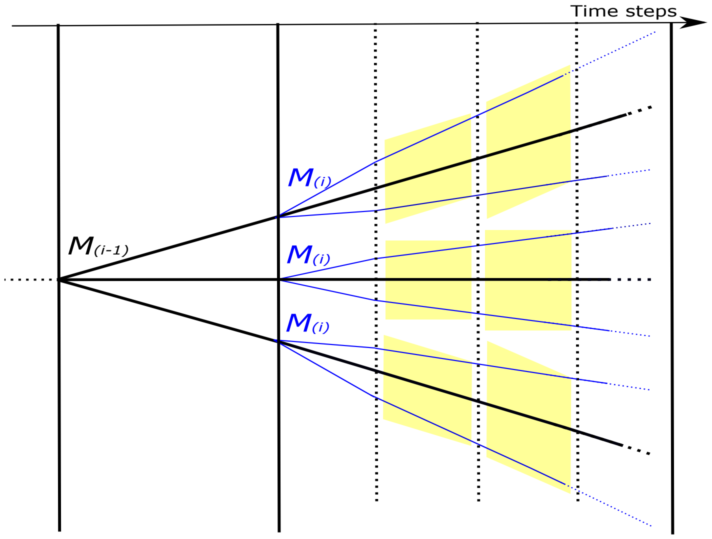


FIGURE 3.8 – Yellow pavings representing inner regressions that need be performed iteratively at decreasing coarse grid times, for instance in the context of the Bermudan put CVA case study of Sect. 3.4.2. The fine blue paths denote inner resimulated paths.

When the systems are well conditioned, the best solutions remain Cholesky or LDL^{T} (i.e. Cholesky implementation avoiding the use of square root functions) factorizations. A simple parallelization strategy would be to execute one resolution per thread. But this is far from optimal. For a system with n unknowns, the most appropriate GPU adaptation of LDL^{T}, presented in [5], involves $1 \leq n^* \leq n$ threads that work together to solve each system. In Figure 3.9 (Left), we see a clear benefit of using $n^* > 1$ threads per system when compared to the straightforward parallelization that involves only one thread per system. In most Longstaff-Schwartz applications, the number of regressors is smaller than 20 and the speedup is ~ 2.

In addition, [5] explain how to filter out the numerically null eigenvalues of an ill-conditioned regression covariance matrix, as the ones that arise in nested regressions based on very few inner trajectories.

3.9.2 Regressions on Outer Trajectories

In contrast to DP on inner trajectories, BSDE simulation on outer trajectories requires only few resolutions (one per coarse time step) of linear systems arising from pricing equations of the form

$$Y_{s_k} = \mathbb{E}\left(\Phi(Z_{s_{k+1}})|Z_{s_k}\right) \tag{3.9.1}$$

at coarse times s_k, where $Z = (X, H)$ includes both market factors X and default indicators encoded in the Markov chain component H (see Sect. 3.3.1). Many outer paths are required for the sake of accuracy, but this is doable without harm provided the regression covariance matrices are estimated off-line based on a large number of outer trajectories. Once the covariance matrices computed, decomposed into eigenelements, and stored in the GPU RAM, they can be reused for any BSDE involving the same model on the outer layer of NMC.

Moreover, once we have the eigenelements of each covariance matrix, principal component analysis (PCA) can be used to reduce the dimensionality of the computations. Figure 3.9 (right) shows the speedup obtained from keeping only 90%

to 95% of the spectrum (total variance) in a linear regression involving a common shock default model driven by three CIR processes ($|X| = 3$) and specifying the default of 100 obligors. As time passes, it becomes more difficult to explain most of the variance with few eigenvalues, because the structure of the likely default configurations becomes richer (as default intensities are "small"). However, in the 90% as in the 95% case, the dimensionality reduction speedup is never below 2, meaning that it is never necessary to use more than half of the eigenvalues, hence half of the time of what a full computation would take, to have a regression accounting for 90% or 95% of the spectrum.

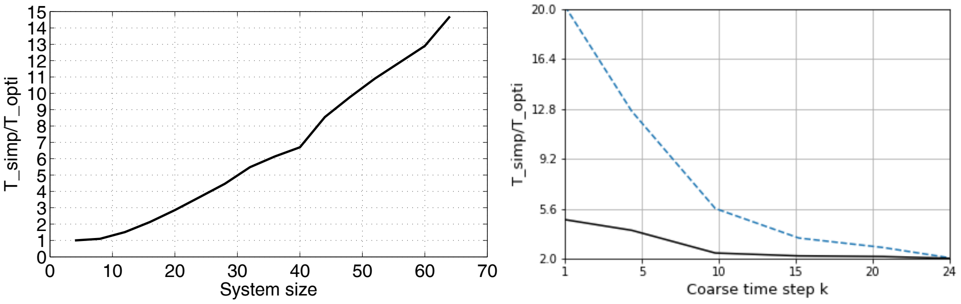


FIGURE 3.9 – The speedup obtained from optimized regressions. Left : in the case of regressions on inner trajectories, such as the ones that appear in the Bermudan put CVA case study of Sect. 3.4.2, by devoting an optimal number $n^* \geq 1$ of threads to each linear system—speedup depending on the size of the systems. Right : in the case of regressions on outer trajectories, such as the ones that appear in the FVA case study of Sect. 3.4.3, by limiting the number of eigenvalues accounted for in the regressions—the '- - ' and '—' curve display the speedup as a function of the coarse time step k when keeping 90%, respectively 95% of the spectrum.

Chapitre 4

Conditional Monte Carlo Learning for diffusions

4.1 Introduction

Numerous contributions in numerical methods based on Monte Carlo reached recently their limits in dealing with the curse of dimensionality [31]. In contrast to previous works, our method is based on a judicious combination between 1NMC and the use of regression. This paper is a natural progress of an increasing interest in NMC started in [63, 68, 86] and used with regression in [2, 37]. Considering a filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_{0\leq t\leq T}, P)$, an $\mathcal{F}_t$-Markov process $(X_t)_{t\in[0,T]}$ taking its values on $\mathbb{R}^{d_1}$ and the time discretization $\{t_0, ..., t_{2^L}\} = \left\{0, T/2^L, ..., T\right\}$, let U_s be a functional of X defined for $s \in \{t_0, ..., t_{2^L}\}$ by

$$(f) \qquad U_s = \mathbb{E}_s\left(\sum_{t_k\geq s}^{t_{2^L}} f(t_k, X_{t_k}, X_{t_{k+1}})\right) = \mathbb{E}\left(\sum_{t_k\geq s}^{T} f(t_k, X_{t_k}, X_{t_{k+1}})\middle|\mathcal{F}_s\right),$$

where $\mathbb{E}_s(\cdot) = \mathbb{E}\left(\cdot\middle|\mathcal{F}_s\right)$, the expectation is always considered under P, each deterministic function $f(t_k, \cdot, \cdot)$ is $\mathcal{B}(\mathbb{R}^{d_1})\otimes\mathcal{B}(\mathbb{R}^{d_1})$-measurable and assumed to satisfy the square integrability $\mathbb{E}(f^2(t_k, X_{t_k}, X_{t_{k+1}})) < \infty$ with convention $f(t_{2^L}, X_{t_{2^L}}, X_{t_{2^L+1}}) = f(t_{2^L}, X_{t_{2^L}})$. The simulation of U is generic to all BSDEs and RBSDEs examples presented in this paper. As nested simulations involve heavy notations, it is easier to present the whole algorithm for the simulation of U then apply it on specific examples.

When previous contributions target estimations of U_{t_k} for $k = 0, ..., 2^L$ knowing some realization of $\{X_{t_j}\}_{0\leq j\leq k}$ $(m_0 = 1, ..., M_0)$, our purpose is to simulate approximations $\{U^{m_0,m_1}_{t_k,s}\}_{s\geq t_{k+1}}$, with $(m_0 = 1, ..., M_0)$ and $(m_1 = 1, ..., M_1)$, of $\{U_s\}_{s\geq t_{k+1}}$ conditionally on the realization $\{X^{m_0}_{t_j}\}_{0\leq j\leq k}$. This task requires the simulation of a first layer $(X^{m_0})_{m_0=1,...,M_0}$ of trajectories that are kept in the machine's memory, then a second unstored layer $(X^{m_0,m_1})_{m_1=1,...,M_1}$ of trajectories, on the top of the first layer, only used to learn how should we perform approximations $\{U^{m_0,m_1}_{t_k,s}\}_{s\geq t_{k+1}}$.Although more complex, this procedure provides much more information on the process U. In particular, we use $\displaystyle\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(f(t_k, X^{m_0}_{t_k}, X^{m_0,m_1}_{t_{k+1}}) + U^{m_0,m_1}_{t_k,t_{k+1}}\right)$ to have the first layer approximation $U^{m_0}_{t_k}$ of U_{t_k}. Knowing the second layer approximation U^{m_0,m_1}, we can compute quantiles on U or, even more remarkable, can simulate another process $\widetilde{U}$ that satisfies equation $(\widetilde{f})$ (Replace f by $\widetilde{f}$ in equation (f)) with an $\widetilde{f}$ that can be a function of U like for instance $\widetilde{f}(t_k, x, y) = f(t_k, U_{t_k}(x), U_{t_{k+1}}(y))$. Consequently, when sufficient assumptions are satisfied, we can learn how to compute functionals of functionals of X with the same 1NMC. This latter fact makes possible a straightforward simulation of Valuation Adjustments [2] as long as one can write them as a composition of functionals then start simulating by the innermost functional till the most outer composition.

Although we are not the first to propose a learning procedure for BSDEs [25], we are the first to do it using nested Monte Carlo instead of a neural network. To the best of our knowledge, we are also first to provide a comprehensive presentation of an iterative algorithm where the accuracy of the estimator $\{U^{m_0}_{t_k}\}_{k=0,\dots,2^L}$ improves by adding more regression steps and thus by increasing the learning depth. Thanks to our method, one can easily balance between complexity and accuracy. Moreover, it is possible to improve the accuracy in a parareal fashion [89] which increases further the parallel scalability of the algorithm. In addition to that, we use equality

$$\mathbb{E}(U_s) = \mathbb{E}\left(U_{s'} + \sum_{t_{l+1}>s}^{s'} f(t_l, X_{t_l}, X_{t_{l+1}})\right)$$

true for $s' > s$, and its localized version for each interval $[a, b]$

$$\mathbb{E}\left(U_s 1_{\{U_s\in[a,b]\}}\right) = \mathbb{E}\left(1_{\{U_s\in[a,b]\}}\left[U_{s'} + \sum_{t_{l+1}>s}^{s'} f(t_l, X_{t_l}, X_{t_{l+1}})\right]\right),$$

to present a nonparametric method to effectively estimate and control the bias. In the same fashion, we detail a variance adjustment procedure based on the equality

$$\mathbb{E}\left(\mathrm{Var}_s(U_{s'})\right) = \mathbb{E}\left(\mathbb{E}_s\left([U_{s'} - \mathbb{E}_s(U_{s'})]^2\right)\right) = \mathbb{E}\left([U_{s'} - \mathbb{E}_s(U_{s'})]^2\right).$$

true for $s' > s$, and its localized version for each interval $[a, b]$

$$\mathbb{E}\left(\mathrm{Var}_s(U_{s'}) 1_{\{\mathrm{Var}_s(U_{s''})\in[a,b]\}}\right) = \mathbb{E}\left(1_{\{\mathrm{Var}_s(U_{s''})\in[a,b]\}}[U_{s'} - \mathbb{E}_s(U_{s'})]^2\right),$$

true for $s' > s$ and $s'' > s$. The proposed variance adjustment strategy makes possible the nested simulation of distribution tails without requiring an importance sampling technique [80]. The good representation of tail events, via variance adjustment, becomes necessary for some nonlinear problems especially RBSDEs. Both bias control and variance adjustment shows that : 1NMC makes possible a very fine tracking of the bias of the first layer fine estimator U^{m_0} and the variance of the second layer coarse estimator U^{m_0,m_1}.

Focusing on the simulation of U given in (f), Section 4.2 introduces the method as well as notations. Section 4.2 also presents the iterative procedure, the bias control and the variance adjustment strategy on the approximation of U. Section 4.3 illustrates the presented method on some standard problems involving BSDEs, RBSDEs and risk measures. These examples show how the algorithm should be adapted to different situations, in particular how to set : iterations, bias control and variance adjustment for BSDEs and optimal stopping problems. Section 4.4 details the required assumptions in a general diffusion setting. It also provides different error estimates associated to our method and gives a sense to the overall approximation procedure. Section 4.5 shows the robustness of our method on highly dimensional problems beyond what is known to be possible in previous papers.

4.2 Conditional learning procedure : Notations and method

In Section 4.2.1, we present the algorithm steps and what should be done to stabilize it. As needed for any learning method, the initialization is also explained in Section 4.2.1. This will set the stage to express, in Section 4.2.2, the regression based approximations as an output of an iterative procedure. Details on the computation of the regression are provided in Section 4.2.3 that also includes a bias control and a variance adjustment necessary when targeting the tail events.

4.2.1 Iterative procedure, regression initialization and stabilization

Using a sufficiently fine discretization $\{t_0, ..., t_{2^L}\} = \{0, \Delta_t, 2\Delta_t, ..., T\}$ with $\Delta_t = T/2^L$, one simulates M_0 realizations $(X^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$ of the Markov process X starting at a deterministic point $X_0 = x_0 \in \mathbb{R}^{d^1}$ with the following induction

$$X^{m_0}_{t_k} = \mathcal{E}_{t_{k-1}}(X^{m_0}_{t_{k-1}}, \xi^{m_0}_{t_k}), \text{ when } k \geq 1 \text{ and } X^{m_0}_{t_0} = x_0, \tag{4.2.1}$$

where $(\xi^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$ are independent realizations of an $\mathbb{R}^{d_2}$ random vector ξ and $(\mathcal{E}_{t_k})_{k=0,...,2^L-1} : \mathbb{R}^{d_1+d_2} \to \mathbb{R}^{d_1}$ are Borel-measurable functions. We use $X^{m_0,1}_{t_k}$, ..., $X^{m_0,d_1}_{t_k}$ to denote the d_1 components of the vector $X^{m_0}_{t_k}$. The sample $(X^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$ stays on the machine memory and is supposed to approximate accurately $(X_t)_{t\in[0,T]}$ in a sense explained in Section 4.4.

For a decreasing sequence $(s_j)_{j=0,...,2^L}$ that takes its values in the time discretization set $\{t_0, ..., t_{2^L}\}$, an extra simulation conditional to the starting $X^{m_0}_{s_j}$ is needed for the learning procedure. Introducing independent realizations $(\xi^{m_0,m_1}_{t_j,t_k})^{(m_0,m_1)\in\{1,...,M_0\}\times\{1,...,M_1+M_1'\}}_{k\in\{j,...,2^L\},j\in\{1,...,2^L\}}$ of the random vector ξ that are also independent from $(\xi^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$, we set for $t_{k-1} \geq s_j$

$$X^{m_0,m_1}_{s_j,t_k} = \mathcal{E}_{t_{k-1}}(X^{m_0,m_1}_{s_j,t_{k-1}}, \xi^{m_0,m_1}_{s_j,t_k}) \text{ and } X^{m_0,m_1}_{s_j,s_j}\Big|_{m_1=1,...,M_1+M_1'} = X^{m_0}_{s_j}. \tag{4.2.2}$$

We use $X^{m_0,m_1,1}_{s_j,t_k}$, ..., $X^{m_0,m_1,d_1}_{s_j,t_k}$ to denote the d_1 components of the vector $X^{m_0,m_1}_{s_j,t_k}$. For $s_j \leq s_l \leq s_k$, we also introduce the notation $X^{m_0,m_1}_{s_j,s_l:s_k}$ and $\xi^{m_0,m_1}_{s_j,s_l:s_k}$ for repectively $(X^{m_0,m_1}_{s_j,s_l}, X^{m_0,m_1}_{s_j,s_l+\Delta_t}, ..., X^{m_0,m_1}_{s_j,s_k-\Delta_t}, X^{m_0,m_1}_{s_j,s_k})$ and $(\xi^{m_0,m_1}_{s_j,s_l}, \xi^{m_0,m_1}_{s_j,s_l+\Delta_t}, ..., \xi^{m_0,m_1}_{s_j,s_k-\Delta}, \xi^{m_0,m_1}_{s_j,s_k})$.

For a positive integer $L' \in]L/2, L]$, the value of each term of the sequence $(s_j)_{j=0,...,2^L}$ is given by its corresponding term in $(T - s^i_j)_{j=0,...,2^L}$ which is defined iteratively for $i = 0, ..., L - L'$ starting with a homogeneously distributed sequence where each term is repeated $2^{L-L'}$ times as follows

$$(s^0_j)_{j=1,...,2^L} = \left\{\frac{T}{2^{L'}}, ..., \frac{T}{2^{L'}}, ..., \frac{(2^{L'}-1)T}{2^{L'}}, ..., \frac{(2^{L'}-1)T}{2^{L'}}, T, ..., T\right\}, \; s^0_0 = 0. \tag{4.2.3}$$

We denote $\mathcal{S}^i$ the set of values taken by $(T - s^i_j)_{j=0,...,2^L}$, for example $\mathcal{S}^0 = \{T, (2^{L'} - 1)T/2^{L'}, ..., T/2^{L'}, 0\}$.

The goal of iterations is to reduce an error term $(e_{T-s^i_j})^{i=0,...,L-L'}_{j=1,...,2^L}$ to make it smaller than some threshold error ε. The expression of the $\mathbb{R}$-valued random processes e and ε will be given in definitions 4.2.1, 4.3.1 and 4.3.2.

We set $j_0^* = s_0^i\Big|_{i=1,...,L-L'} = \max(\emptyset) = 0$, for each step $i = 1, ..., L - L'$ we define $Q_i = 2^{L-L'-i}$ and $(\widehat{s}_j^{i-1})_{j=0,...,2^L}$

$$\left\{\begin{array}{c} \text{When } j \leq j_{i-1}^* \text{ define } \widehat{s}_j^{i-1} = s_j^{i-1} \\ \text{Otherwise, for } j' > j_{i-1}^*/Q_i \text{ set } \widehat{s}_j^i\Big|_{j=Q_i(j'-1)+1}^{Q_ij'} = \dfrac{s_{Q_ij'}^{i-1} + s_{Q_i(j'-1)}^{i-1}}{2}, \end{array}\right. \tag{4.2.4}$$

and we denote $\widehat{\mathcal{S}}^{i-1}$ the set of values taken by $(T - \widehat{s}_j^{i-1})_{j=0,...,2^L}$. Then, we consider the following actualization strategy :

1 Compute $\left(e_{T-s_j^{i-1}}^{\mathcal{S}^{i-1}}\right)_{j>j_{i-1}^*}$

2 Use $j_i^* = j_{i-1}^* \vee \max\left(\left\{j > j_{i-1}^*;\ e_{T-s_k^{i-1}}^{\mathcal{S}^{i-1}} < \varepsilon_{T-s_k^{i-1}}^{\mathcal{S}^{i-1}} \text{ for } k \leq j\right\}\right)$ with $x \vee y = \max(x, y)$ to define

$$s_j^i = s_j^{i-1} 1_{j \leq j_i^*} + \widehat{s}_j^{i-1} 1_{j > j_i^*}. \tag{4.2.5}$$

The notation $s_j^i\Big|_{j=Q_i(j'-1)+1}^{Q_ij'}$ is used for $s_{Q_i(j'-1)+1}^i, ..., s_{Q_ij'}^i$. In Figure 4.1, we illustrate how this discretization strategy is implemented, in particular we chose $L' > L/2$.

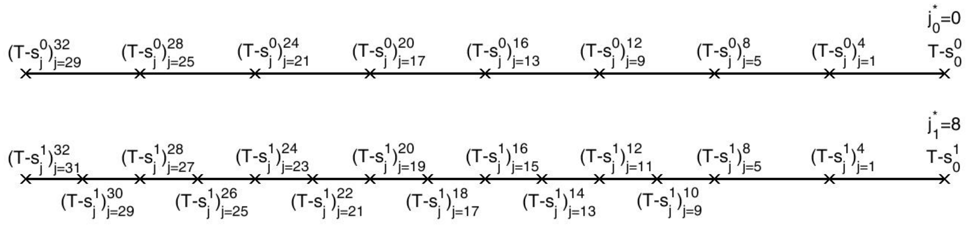


FIGURE 4.1 – An example for (4.2.5) when $i = 0, 1$, $L = 5$ and $L' = 3$.

Remark 4.2.1 *Expression* (4.2.5) *ensures that $s_{2^L}^i$ is always equal to T. Thus $s_{2^L} = 0$ which will be involved in definitions 4.2.1, 4.3.1 and 4.3.2 to introduce both a simulated value and an average on learned values at time* 0*.*

In (4.2.2), we simulate $M_1 + M_1'$ conditional realizations of X in order to keep those indexed from $m_1 = M_1+1$ to $m_1 = M_1+M_1'$ for the approximation of regression matrices. Consequently, we made explicit the independence between trajectories used for the estimation of regression matrices and those used in the backward induction. To reduce the complexity of the algorithm and memory occupation, trajectories used for regression matrices can be simulated offline then erased from the memory. Given m_0, if an inner trajectory from $\{X^{m_0,m_1}\}_{m_1=1,...,M_1}$ is needed α times in the backward induction, we simulate α independent copies of it and use each copy once. This reduces further memory occupation as well as any superfluous dependence structure.

For each ordered couple $(j < k)$ of indices that take their values in $\{1, ..., 2^L\}$, we introduce a stabilization operator

$$\mathcal{T}_{t_j,t_k,M_1'}^{m_0} : \mathbb{R}^{d_1} \ni x \mapsto {}^t\widetilde{\Gamma}_{t_j,t_k,M_1'}^{m_0}\left(x - X_{t_k}^{m_0}\right) \in \mathbb{R}^{d_1'}\ (d_1' \leq d_1), \tag{4.2.6}$$

that performs a linear combination of the components of $\left(x - X_{t_k}^{m_0}\right)$ using $\widetilde{\Gamma}_{t_j,t_k,M_1'}^{m_0}$ that contains some eigenvectors from $\Gamma_{t_j,t_k,M_1'}^{m_0}$ obtained with the eigenvalue decomposition

$$\Gamma_{t_j,t_k,M_1'}^{m_0}\Lambda_{t_j,t_k,M_1'}^{m_0}\,{}^{t}\Gamma_{t_j,t_k,M_1'}^{m_0} \tag{4.2.7}$$

of the regression matrix

$$\frac{1}{M_1'}\sum_{m_1=M_1+1}^{M_1+M_1'}\left(X_{t_j,t_k}^{m_0,m_1} - X_{t_k}^{m_0}\right){}^{t}\left(X_{t_j,t_k}^{m_0,m_1} - X_{t_k}^{m_0}\right) \tag{4.2.8}$$

where ^{t} is the transposition operation.

Once factorization (4.2.7)=(4.2.8) is performed, we obtain the diagonal matrix $\Lambda_{t_j,t_k,M_1'}^{m_0} = \operatorname{diag}\left(\left\{\lambda_{t_j,t_k,M_1'}^{m_0,l}\right\}_{l=1,\ldots,d_1}\right)$ of decreasing positive eigenvalues. Then, we define $\widetilde{\Lambda}_{t_j,t_k,M_1'}^{m_0} = \operatorname{diag}\left(\left\{\lambda_{t_j,t_k,M_1'}^{m_0,l}\right\}_{l=1,\ldots,d_1'}\right)$ as the truncation of $\Lambda_{t_j,t_k,M_1'}^{m_0}$ with d_1' defined by

$$d_1' = \min\left\{k \in \{1,..,d_1''\},\ \sum_{l=1}^{k}\lambda_{t_j,t_k,M_1'}^{m_0,l} \geq 95\%\sum_{l=1}^{d_1''}\lambda_{t_j,t_k,M_1'}^{m_0,l}\right\}, \tag{4.2.9}$$

where d_1'' keeps only eigenvalues that make the regression problem well-conditioned i.e. The ratio $\left.\dfrac{\lambda_{t_j,t_k,M_1'}^{m_0,l}}{\lambda_{t_j,t_k,M_1'}^{m_0,1}}\right|_{l=1,\ldots,d_1''}$ has to be bigger than 10^{-6} in single precision or bigger than 10^{-15} in double precision floating representation [103]. In addition to ensuring a well-conditioned regression problem, equality (4.2.9) also performs a principal component analysis [103]. At the same time that we set the components of $\widetilde{\Lambda}_{t_j,t_k,M_1'}^{m_0}$, we define the matrix $\widetilde{\Gamma}_{t_j,t_k,M_1'}^{m_0}$ that contains only the eigenvectors in $\Gamma_{t_j,t_k,M_1'}^{m_0}$ that are associated to $\widetilde{\Lambda}_{t_j,t_k,M_1'}^{m_0}$.

Regressing with respect to ${}^{t}\widetilde{\Gamma}_{t_j,t_k,M_1'}^{m_0}\left(X_{t_j,t_k}^{m_0,m_1} - X_{t_k}^{m_0}\right) \in \mathbb{R}^{d_1'}$ ($d_1' \leq d_1$), instead of $\left(X_{t_j,t_k}^{m_0,m_1} - X_{t_k}^{m_0}\right) \in \mathbb{R}^{d_1}$, involves the inversion of the diagonal matrix $\widetilde{\Lambda}_{t_j,t_k,M_1'}^{m_0}$ which replaces the whole regression matrix (4.2.8). Since $\widetilde{\Lambda}$ is bounded below away from zero, its inverse is bounded and the same for the regression procedure. This stabilizes the computation of the regression estimator whose expression is detailed in Section 4.2.3. Besides, since we have a large number of regression matrices, we can batch compute these inversions like explained in [5].

For $t_0 \leq s_j < s_k < T$ and conditionally to $X_{s_j}^{m_0}$, we want to keep only first/low order regression terms around $X_{s_k}^{m_0}$. We also want to reduce the bias induced by successive regressions as explained in Section 4.2.3. A natural way to do this is to make sure that the time distance $s_k - s_j$ is sufficiently small to neglect the higher order terms as well as to reduce bias propagation between s_k and s_j. For this purpose, we appropriately initialize the value L' ($> L/2$) as well as the couple $(\underline{s_j}, \overline{s_j})$ then at each iteration i we actualize the value taken by this couple according to (4.2.25) and (4.2.26).

When $i = 0$, for each $j = 0, ..., 2^L$, we define $\overline{s_j^{\mathcal{S}^0}}(s_j^0)$ as

$$\overline{s_j^{\mathcal{S}^0}}(s_j^0)=\max\left\{\begin{matrix} u \in \mathcal{S}^0\cap]T- s_j^0, \overline{\delta(T- s_j^0)}]; \text{(Bias Control) satisfied at} \\ T - s_j^0 \text{ and (4.2.11) fulfilled for all } s \in \mathcal{S}^0\cap]T - s_j^0, u] \end{matrix}\right\} \quad (4.2.10)$$

with δ defined in (4.2.14), to simplify the understanding we can start assuming $\overline{\delta(T- s_j^0)} = T$. The inteval $]T- s_j^0, \overline{\delta(T- s_j^0)}]$ will be better specified at each definition 4.2.1, 4.3.1 and 4.3.2.

Then $J_j^0 = \mathcal{S}^0 \cap \left] s_j, \overline{s_j^{\mathcal{S}^0}}(s_j^0)\right]$ is a set of strictly decreasing time increments with the control (Bias Control), specified in definitions 4.2.1, 4.3.1 and 4.3.2, that also satisfy

$$\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\sum_{l=1}^{d_1}\frac{1}{M_1'}\sum_{m_1=M_1+1}^{M_1+M_1'}\left(X_{s_j,s}^{m_0,m_1,l}- X_s^{m_0,l}\right)^2\right) < \epsilon_{1,s_j} \quad (4.2.11)$$

for some tuning positive parameter ϵ_{1,s_j}. We consequently initialize

$$\overline{s_j^{\mathcal{S}^0}}(s_j^0) = \max(J_j^0),\ \underline{s_j^{\mathcal{S}^0}}(s_j^0) = \min(J_j^0) \text{ and } J_j^0 = \mathcal{S}^0 \cap \left] s_j, \overline{s_j^{\mathcal{S}^0}}(s_j^0)\right] \quad (4.2.12)$$

In what follows, if iteration index i is set and there is no confusion on the chosen set $\mathcal{S}^i$, we simplify notations and use $\overline{s_j}$ and $\underline{s_j}$ instead of $\overline{s_j^{\mathcal{S}^i}}(s_j^i)$ and $\underline{s_j^{\mathcal{S}^i}}(s_j^i)$.

By definition, J_j^0 contains different elements and we use $|J_j^0|$ to denote its cardinal. For any j such that $\overline{s_j} < T$, we choose the right values for L' to ensure that $2^{L-L'} < |J_j^0| \leq 2^{L'}$. Consequently, at the initialization step, one increases progressively L' till the latter condition is fulfilled. When $i > 0$ and $j = 0, ..., 2^L$, we define

$$J_j^i = \mathcal{S}^i\cap]s_j, \overline{s_j^{\mathcal{S}^i}}(s_j^i)]. \quad (4.2.13)$$

Given that $(s_j)_{j\in\{0,...,2^L\}}$ is a decreasing, and not strictly decreasing, sequence of coarse increments, we need to define on $\mathcal{S}^i$ a new operator $\delta^{\mathcal{S}^i}$ that associates to each $s \in \mathcal{S}^i$ the next increment in $\mathcal{S}^i$. For a fixed index $j \in \{1, ..., 2^L\}$, we define $\delta_{s_j}^{\mathcal{S}^i}(\cdot)$ on $(s_k)_{k\leq j}$, taken its values in $\mathcal{S}^i \cap [s_j, \overline{s_j}]$ $(= \{s_j\} \cup J_j^i)$, by

$$\delta_{s_j}^{\mathcal{S}^i}(s_k) = \min\left(\overline{s_j}, \min\{s \in \mathcal{S}^i; s_k < s \leq \overline{s_j}\}\right) \quad (4.2.14)$$

with $\min(\emptyset) = \infty$.

When there is no confusion on the chosen set $\mathcal{S}^i$, we use δ_{s_j} notation instead of $\delta_{s_j}^{\mathcal{S}^i}$. When $s_k < \overline{s_j}$, we use $\delta^{\mathcal{S}^i}$ notation instead of $\delta_{s_j}^{\mathcal{S}^i}$. When there is no confusion on the chosen set $\mathcal{S}^i$ and $s_k < \overline{s_j}$, we simplify both indices and use δ instead of $\delta_{s_j}^{\mathcal{S}^i}$.

This time operator will be largely used and for a given set $\mathcal{S}^i$ it has the following properties

Pr1. (4.2.26) makes $\underline{s_j} = \delta_{s_j}(s_j) = \delta(s_j)$.

Pr2. As long as $\max(s_{j_1}, s_{j_2}) \leq s_k < \min(\overline{s_{j_1}}, \overline{s_{j_2}})$, $\delta_{s_{j_1}}(s_k) = \delta_{s_{j_2}}(s_k) = \delta(s_k)$.

Pr3. For fixed iteration step i, the nth composition of δ_{s_j} denoted $\delta_{s_j}^n(\cdot)$ is equal to $\overline{s_j}$ when $n \geq |J_j^i|$.

4.2.2 Fine and coarse approximations

Based on what was presented in Section 4.2.1, we detail here the simulation of approximations of U defined by (f). Considering the discretization sequence $(s_j)_{j=0,...,2^L}$ that takes its values in the set $\mathcal{S} \subset \{t_0, ..., t_{2^L}\}$, we use a learning procedure to associate to each scenario m_0 and each discretization set $\mathcal{S}$ a couple of function families $(\widetilde{h}^{m_0,\mathcal{S}}, \overline{h}^{m_0,\mathcal{S}})$.

Now, for given indices $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j}$, for $x \in \mathbb{R}^{d_1}$ and $s \in \{s_k, s_k + \Delta_t, ..., \delta(s_k) - \Delta_t\}$, we define two approximation levels : A coarse approximation around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ defined by

$$\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \ell\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) H^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{4.2.15}$$

and a fine approximation at $X^{m_0}_s$ defined by

$$\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\mathcal{S}}_{s_k,\delta(s_k)}(X^{m_0,m_1}_{s,\delta(s_k)}) + \sum_{t_{l+1}>s}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})\right]. \tag{4.2.16}$$

To complete this inductive interconnected backward definition of $\overline{h}$ and $\widetilde{h}$, we set the final coarse approximation to

$$\overline{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}(x) = \begin{cases} f(T,x) & \text{if } \overline{s_j} = T, \\ \overline{h}^{m_0,\mathcal{S}}_{\underline{s_j},\overline{s_j}}(x) = \overline{h}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{s_j}}(x) & \text{if } \overline{s_j} < T, \end{cases} \tag{4.2.17}$$

where $\overline{s_j} > \underline{s_j} > s_j$ are specified during the initialization phase (cf. (4.2.12)) then actualized at each step (cf. (4.2.25) and (4.2.26)). $\overline{s_j}$ and $\underline{s_j}$ are really needed when T is sufficiently big or the variance produced by X is large enough. Otherwise, (4.2.17) can be replaced by $\overline{h}^{m_0,\mathcal{S}}_{s_j,T}(x) = f(t_{2^L}, x) = f(T, x)$.

$\mathcal{T}$ involved in (4.2.15) was already defined in (4.2.6). The value of the regression constant $\ell\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\Big]$ depends on the variance adjustment procedure presented in section 4.2.3. However, the straight implementation can simply set $\ell\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] = \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$ for any couple (s_j, s_k) satisfying $s_j < s_k \leq \overline{s_j}$. Regarding the regression vector $H^{m_0,\mathcal{S}}_{s_j,s_k}$, its value is obtained from an estimation of the vector $a \in \mathbb{R}^{d_1'}$ that minimizes the quadratic error given by

$$\mathbb{E}\left[\mathbb{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)}) - {}^t a\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\right]^2 \tag{4.2.18}$$

with $X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)} = \Big(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,s_k+\Delta_t}, ..., X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)-\Delta_t}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}\Big)$ and

$$\mathbb{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x) = \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}\left(x_{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}}\right) - \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + \sum_{l=1}^{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}-1} f(t_{k_{s_k}+l}, x_l, x_{l+1}) \tag{4.2.19}$$

where $k_{s_k} = s_k/\Delta_t - 1$, $x = (x_1, ..., x_{(\delta_{s_j}(s_k)-s_k)/\Delta_t})$ with each coordinate of x belonging to $\mathbb{R}^{d_1}$.

When $s_k = \delta_{s_j}(s_j)$ and $\ell\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] = \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$, one can check the coherence of the previous definitions aimed to approximate U defined in (f). Indeed, (4.2.16) would provide for any $s \in \{s_j, s_j + \Delta_t, ..., \delta(s_j) - \Delta_t\} = \{s_j, s_j + \Delta_t, ..., s_k - \Delta_t\}$

$$\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_j}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) + \sum_{t_{l+1}>s}^{s_k} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})\right] \qquad (4.2.20)$$

where the term $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}$, defined in (4.2.15), is obtained through the projection of $\sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}) + \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)})$, involved in (4.2.30), on $\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})$. In addition, if we had $\delta^2(s_j) = \delta(s_k) = \overline{s_j} = t_{2^L} = T$ then (4.2.17) would make $\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(\cdot) = f(t_{2^L}, \cdot)$ and as $s \in \{s_j, s_j + \Delta_t, ..., s_k - \Delta_t\}$ the definition of $\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_j}}$ would involve $\sum_{t_{l+1}>s}^{s_k} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$ plus the projection of $\sum_{t_{l+1}>s_k}^{t_{2^L}} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}) + f(T, X^{m_0,m_1}_{s_j,T})$ as shown on Figure 4.2. $\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$ is equal to $\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\sum_{t_{l+1}>s_k}^{t_{2^L}} f(t_l, X^{m_0,m_1}_{s_k,t_l}, X^{m_0,m_1}_{s_k,t_{l+1}}) + f(T, X^{m_0,m_1}_{s_k,T})\right]$ because $\delta(s_k) = T$.

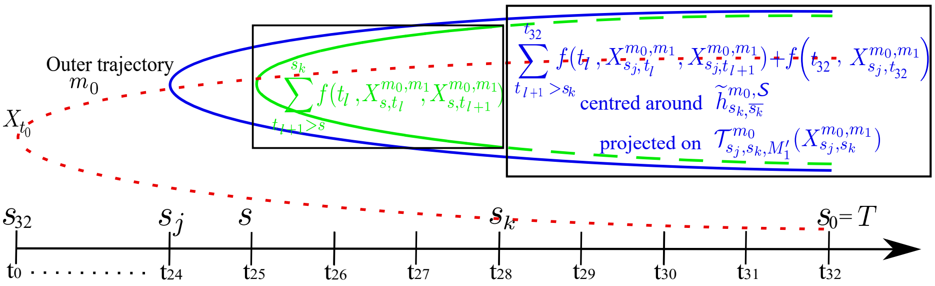

Outer trajectory
m_0
X_{t_0}
$\sum_{t_{l+1}>s}^{s_k} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$
$\sum_{t_{l+1}>s_k}^{t_{32}} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}) + f\Big(t_{32}, X^{m_0,m_1}_{s_j,t_{32}}\Big)$
centred around $\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$
projected on $\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})$
s_{32}
s_j
s
s_k
$s_0 = T$
t_0 ....
t_{24}
t_{25}
t_{26}
t_{27}
t_{28}
t_{29}
t_{30}
t_{31}
t_{32}

FIGURE 4.2 – An example for (4.2.20) when $\delta^2(s_j) = \delta(s_k) = \overline{s_j} = t_{2^L} = T$, $L = 5$ and $L' = 3$.

According to equations (4.2.15), (4.2.16), (4.2.18) and (4.2.19), the functions $\overline{h}$ and $\widetilde{h}$ are defined backwardly. When $\widetilde{h}$ is a straight Monte Carlo involving $\overline{h}$, the latter is defined using a regression around a point at which we expressed $\widetilde{h}$. Consequently, $\overline{h}$ can be seen as a conditional first order Taylor expansion around the first layer of trajectories $(X^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$. The term of order zero in this expansion is played by $\ell[\cdot]$, where the term ${}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x)H^{m_0,\mathcal{S}}_{s_j,s_k}$, deduced from the minimization of (4.2.18), plays the order one.

Remark 4.2.2 *1. Since we do not want to increase further the algorithm complexity by considering higher order terms, the definition of $\overline{h}$ involves only linear regression around $X^{m_0}_{s_k}$.*

2. When the dimension d_1 is not too high, it is possible to regress the residual of the first regression on higher order terms. These successive regressions do not increase drastically the complexity when compared to the standard procedure. Nevertheless, as it separates regression with respect to first order terms and regression with respect to higher order terms, it loses orthogonality between first and higher order terms.

3. *In case X is a martingale, the linearity simplifies further computations since, for instance,* (4.2.16) *can be replaced by*

$$\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \overline{h}^{m_0,\mathcal{S}}_{s_k,\delta(s_k)}(X^{m_0}_s) + \frac{1}{M_1}\sum_{m_1=1}^{M_1}\sum_{t_{l+1}>s}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}}).$$

Definition 4.2.1 *For* $i^* = \min(\min\{i=1,...,L-L', j^*_i = 2^L\}, L-L')$

— *For* $k < j \in \{1,...,2^L\}$ *that satisfy* $s_j < s_k \leq \overline{s_j} < t_{2^L} = T$, *the simulation* $U^{m_0,m_1}_{s_j,s_k}$ *of* U *around* $X^{m_0}_{s_k}$ *conditionally on* $X^{m_0}_{s_j}$ *is set to be equal to* $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ *where* $\overline{h}$ *is given in* (4.2.15) *and* (4.2.17).

— *For* $k \in \{1,...,2^L\}$ *and* $s \in \{s_k, s_k+\Delta_t, ..., \delta(s_k)-\Delta_t\} - \{0\}$, *the simulation* $U^{m_0}_s$ *of* U *at* $X^{m_0}_s$ *is set to be equal to* $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s,\overline{s_k}}$ *with* $\widetilde{h}$ *expressed in* (4.2.16).

— *The average* U^{lear}_0 *of learned values on* U_0 *is equal to*

$$U^{lear}_0 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}} \tag{4.2.21}$$

and the simulated value U^{sim}_0 *of* U_0 *is equal to*

$$U^{sim}_0 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}} + \sum_{t_{l+1}>0}^{\delta(0)} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right] \tag{4.2.22}$$

with $\widetilde{h}$ *expressed in* (4.2.16).

— *Introduced in* $\underline{(4.2.10)}$, *(Bias Control) associated to* (f) *is defined at* $s \in \mathcal{S}^0$ *for* $u \in \mathcal{S}^0 \cap]s, \overline{\delta(s)}]$ *by*

$$\left|\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{s,u} - \widetilde{h}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}} - \sum_{t_{l+1}>s}^{\delta(s)} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right)\right| < \epsilon^{\mathcal{S}^0}_{2,s}$$

where for each set $\mathcal{S}$, $\{\epsilon^{\mathcal{S}}_{2,s}\}_{s\in\mathcal{S}}$ *is a family of positive bias tuning parameters.*

— *For* $k \in \{j^*_i+1, ..., 2^L\}$, *setting* $s_k = T - s^i_k$ *and noticing that* $\delta^{\mathcal{S}^i}(s_k) = \delta^{\widehat{\mathcal{S}}^i}(\delta^{\widehat{\mathcal{S}}^i}(s_k))$, $e^{\mathcal{S}^i}_{s_k}$ *and* $\varepsilon^{\mathcal{S}^i}_{s_k}$ *are given by*

$$e^{\mathcal{S}^i}_{s_k} = \frac{1}{M_0 M_1}\sum_{m_0=1}^{M_0}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\widehat{\mathcal{S}}^i}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}) - \overline{h}^{m_0,\mathcal{S}^i}_{s_k,\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)})\right],$$

$$\varepsilon^{\mathcal{S}^i}_{s_k} = \sum_{s\in\mathcal{S}^i, s>s_k}\epsilon^{\mathcal{S}^i}_{2,s}.$$

Remark 4.2.3 1. *U^{m_0,m_1} can be seen as the inner or second layer approximation of U and U^{m_0} can be seen as the outer or first layer approximation of U.*

2. *When U^{m_0,m_1} is only defined on $\mathcal{S}^{i^*}$, it is remarkable to see that U^{m_0} is defined on the whole fine discretization set $\{t_0,...,t_{2^L}\}$.*
3. *For any $\mathcal{S}$, it is natural to have $\epsilon^{\mathcal{S}}_{2,s}$ proportional to the value of the estimation $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}}$. Used to control the bias, the choice of $\epsilon^{\mathcal{S}}_{2,s}$ has also to take into account the confidence interval of the estimator of the left side of inequality (Bias Control).*
4. *Although (Bias Control) is quite sufficient to have almost unbiased estimates, Section 4.2.3 introduces a more stringent local bias control.*
5. *$e^{\mathcal{S}^i}$ is defined as the average difference between the estimation $\overline{h}^{m_0,\mathcal{S}^i}$ that involves the discretization set $\mathcal{S}^i$ and the estimation $\overline{h}^{m_0,\widehat{\mathcal{S}}^i}$ that involves a finer discretization set $\widehat{\mathcal{S}}^i$ defined below* (4.2.4). *With actualization* (4.2.5), *we are basically saying that the discretization set should be finer only when the difference between approximations is superior to the sum of possible accumulation of bias $\varepsilon^{\mathcal{S}^i}$.*

4.2.3 Regression computations : Bias control and variance adjustment

As a continuation to Section 4.2.1, we explain the (Bias Control) expression and how the value of $(\underline{s_j},\overline{s_j})$ should be actualized. Then, as a continuation to Section 4.2.2, for each couple (scenario/discretization set) $=(m_0,\mathcal{S})$ we provide possible values of the couple $\left(\ell\left[\overline{h}^{m_0,\mathcal{S}}\right],H^{m_0,\mathcal{S}}\right)$ including a variance adjustment procedure. We remind that both procedures, explained in this section, are only feasible because of the nested nature of our simulation and they would not be possible otherwise.

In Section 4.2.1 equation (4.2.12), we defined $(\underline{s},\overline{s})$ on the discretization set $\mathcal{S}^0$. In order to reduce the backward bias propagation, this definition used the double layer Monte Carlo to control the average bias. Indeed, as $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,u}$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{\delta(s_j),\overline{\delta(s_j)}}+\sum_{t_{l+1}>s_j}^{\delta(s_j)}f(t_l,X^{m_0}_{t_l},X^{m_0}_{t_{l+1}})\right)$ are both approximations of $\mathbb{E}(U_{s_j})=\mathbb{E}\left(U_{\delta(s_j)}+\sum_{t_{l+1}>s_j}^{\delta(s_j)}f(t_l,X^{m_0}_{t_l},X^{m_0}_{t_{l+1}})\right)$, it is natural to have them almost equal. For large values of M_0, the difference between these approximations is due to bias. As explained at the end of Section 4.4.1 and the beginning of Section 4.4.2, a judicious method to reduce this bias propagation is to adjust the number of successive regressions through the appropriate choice of u.

The choice of u in (Bias Control) ought to decrease the global average value of the bias. More local approach can be developed using equality

$$\mathbb{E}\Big(U_{s_j}1_{\{U_{s_j}\in[a,b]\}}\Big)=\mathbb{E}\left(1_{\{U_{s_j}\in[a,b]\}}\left[U_{\delta(s_j)}+\sum_{t_{l+1}>s_j}^{\delta(s_j)}f(t_l,X^{m_0}_{t_l},X^{m_0}_{t_{l+1}})\right]\right)\tag{4.2.23}$$

which is true for any localizing interval $[a,b]$. When M_0 is sufficiently large, one can sort $\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\}_{m_0\leq M_0}$ and define a subdivision of localizing intervals $\{[a_q,a_{q+1}]\}_{q\geq 1}$ then choose $\overline{s_j}$ that does not induce a large difference between $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(1_{\left\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\in[a_q,a_{q+1}]\right\}}\left[\widetilde{h}^{m_0,\mathcal{S}^0}_{\delta(s_j),\overline{\delta(s_j)}}+\sum_{t_{l+1}>s_j}^{\delta(s_j)}f(t_l,X^{m_0}_{t_l},X^{m_0}_{t_{l+1}})\right]\right)$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}1_{\left\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\in[a_q,a_{q+1}]\right\}}\right)$ for any q. This local increase of bias can be even tracked for any $s \in \mathcal{S}_0 \cap [\delta(s_j), \overline{s_j}[$ using the difference

$$\frac{1}{M_0}\sum_{m_0=1}^{M_0} 1_{\left\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}} \in [a_q,a_{q+1}]\right\}}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}} - \widetilde{h}^{m_0,\mathcal{S}^0}_{s,\overline{s}} - \sum_{t_{l+1}>s_j}^{s} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right). \qquad (4.2.24)$$

Although the local tracking of bias was not necessary in our simulations, it is quite remarkable to point out the strength of bias control induced by 1NMC.

For $j = 0, ..., 2^L$ and $s_j = T - s^i_j \in \mathcal{S}^i$, the actualization of $(\underline{s_j}, \overline{s_j})$ is given by

$$\overline{s^{\mathcal{S}^i}_j}(s^i_j) = \overline{s^{\mathcal{S}^{i-1}}_j}(s^{i-1}_j)1_{\overline{I_{i,j}}} + \max\left(\mathcal{S}^i \cap]s_j, \overline{s^{\mathcal{S}^{i-1}}_j}(s^{i-1}_j)[\right) 1_{\overline{I_{i,j}}^c}, \qquad (4.2.25)$$

$$\underline{s^{\mathcal{S}^i}_j}(s^i_j) = \underline{s^{\mathcal{S}^{i-1}}_j}(s^{i-1}_j)1_{\underline{I_{i,j}}} + \min\left(\mathcal{S}^i \cap]s_j, \overline{s^{\mathcal{S}^{i-1}}_j}(s^{i-1}_j)[\right) 1_{\underline{I_{i,j}}^c} \qquad (4.2.26)$$

where the sets of indices $\underline{I_{i,j}} = \{j \leq j^*_i\} \cup \{s^i_j \neq s^{i-1}_j\}$ and $\overline{I_{i,j}} = \underline{I_{i,j}} \cup \{\overline{s^{\mathcal{S}^{i-1}}_j}(s^{i-1}_j) = T\}$. In Figure 4.3, we illustrate what happens when $j > j^*_1$ with either $s^1_j \neq s^0_j$ ($j = 25, 26$) or $s^1_j = s^0_j$ ($j = 27, 28$). Except when $\{\overline{s^{\mathcal{S}^{i-1}}_j}(s^{i-1}_j) = T\}$, the actualization strategy given by equations (4.2.25) and (4.2.26) aims at ensuring $\overline{s^{\mathcal{S}^i}_{j_1}}(s^i_{j_1}) \neq \overline{s^{\mathcal{S}^i}_{j_2}}(s^i_{j_2})$ and $\underline{s^{\mathcal{S}^i}_{j_1}}(s^i_{j_1}) \neq \underline{s^{\mathcal{S}^i}_{j_2}}(s^i_{j_2})$ as long as $s^i_{j_1} \neq s^i_{j_2}$. As mentioned before, if iteration index i is set and there is no confusion on the chosen set $\mathcal{S}^i$, we simplify notations and use $\overline{s_j}$ and $\underline{s_j}$ instead of $\overline{s^{\mathcal{S}^i}_j}(s^i_j)$ and $\underline{s^{\mathcal{S}^i}_j}(s^i_j)$.

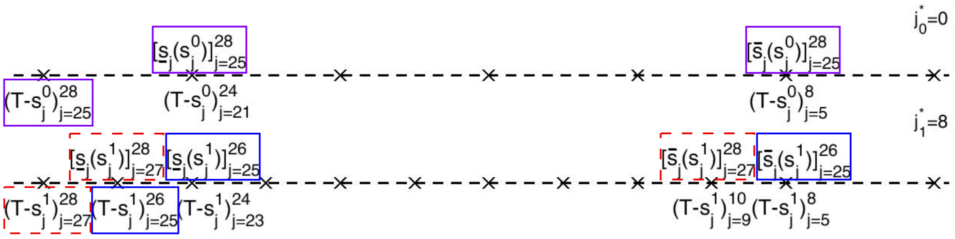


FIGURE 4.3 – An example for (4.2.12), (4.2.25) and (4.2.26) based on the example of Figure 4.1.

Given two indices $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j}$, the expression of $(\ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right], H^{m_0,\mathcal{S}}_{s_j,s_k})$ involves the use of an intermediary variable $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$ and an intermediary vector $\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k}$. Given the value of the couple $(\gamma^{m_0,\mathcal{S}}_{s_j,s_k}, \widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k})$ specified in (4.2.29) and (4.2.32), we define

$$H^{m_0,\mathcal{S}}_{s_j,s_k} = \gamma^{m_0,\mathcal{S}}_{s_j,s_k}\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k} \qquad (4.2.27)$$

and

$$\ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right] = \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + \frac{(1-\gamma^{m_0,\mathcal{S}}_{s_j,s_k})}{M_1}\sum_{m_1=1}^{M_1} {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k}. \qquad (4.2.28)$$

Then, $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$ is used to adjust the variance of $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}$ defined in (4.2.15) without changing its average value. Indeed, the expression of $\ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right]$ makes $\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})/M_1$ invariable with respect to $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$.

The value of $\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k}$ is given by

$$\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k} = (\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1} \frac{1}{M_1} \sum_{m_1=1}^{M_1} \mathcal{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)}) \tag{4.2.29}$$

where $X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)} = \left(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,s_k+\Delta_t}, ..., X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)-\Delta_t}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}\right)$ and the function $\mathcal{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'} : \Omega \times \mathbb{R}^{d_1(\delta_{s_j}(s_k)-s_k)/\Delta_t} \ni (\nu, x_1, ..., x_{(\delta_{s_j}(s_k)-s_k)/\Delta_t}) \to \Omega \times \mathbb{R}^{d_1'}$ is $\mathcal{F}_{\delta_{s_j}(s_k)} \otimes \mathcal{B}(\mathbb{R}^{d_1(\delta_{s_j}(s_k)-s_k)/\Delta_t})$-measurable and defined by

$$\mathcal{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x_1) \underbrace{\begin{bmatrix} \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)} \left(x_{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}}\right) - \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\delta_{s_k}(s_k)} \\ + \displaystyle\sum_{l=1}^{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}-1} f(t_{k_{s_k}+l}, x_l, x_{l+1}) \end{bmatrix}}_{\mathbb{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x)} \tag{4.2.30}$$

where $k_{s_k} = s_k/\Delta_t - 1$ and $x = (x_1, ..., x_{(\delta_{s_j}(s_k)-s_k)/\Delta_t})$.

Regarding $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$, various values can be considered. The straight choice is to take $\gamma^{m_0,\mathcal{S}}_{s_j,s_k} = 1$ which reduces the procedure to a standard regression. However, this is not the suitable choice for problems that heavily depend on tail distribution. Indeed, given two arbitrary square integrable random variables χ_1 and χ_2, consider χ_3 to be the regression of χ_1 with respect to χ_2. Because generally regression preserves the mean value, it is reasonable to assume $\mathbb{E}(\chi_3) = \mathbb{E}(\chi_1)$. However, regressions decreases the second moment i.e. $\mathbb{E}(\chi_3^2) < \mathbb{E}(\chi_1^2)$ and thus $Var(\chi_3) < Var(\chi_1)$. The latter fact becomes a real problem for tail distribution when $Var(\chi_3) << Var(\mathbb{E}(\chi_1|\chi_2))$. Some contributions tackle rare event simulation using a change of probability trick [38, 62] and more recent contribution [6] implements reversible shaking transformations.

In (Bias Control), we established strong constraints to make $\widetilde{h}^{m_0,\mathcal{S}}$ an almost unbiased estimator of U. It is then possible to use their values to propose an appropriate adjustment of the variance. For $s_j < s$ with $s = s_k, \delta(s_k)$, the whole idea is based on the following equality

$$\mathbb{E}\left(\mathrm{Var}_{s_j}(U_s)\right) = \mathbb{E}\left(\mathbb{E}_{s_j}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right)\right) = \mathbb{E}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right).$$

Defining $(\sigma^{\mathcal{S}}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}} - \frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})\right]^2$ and $(\sigma^{m_0,\mathcal{S}}_{s_j,s})^2 = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s}) - \frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})\right]^2$ as the estimators of $\mathbb{E}_{s_j}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right)$ and $\mathbb{E}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right)$ respectively, it is then natural to have for $s = s_k, \delta(s_k)$ as M_1 and $M_0 \to \infty$

$$(\sigma^{\mathcal{S}}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}(\sigma^{m_0,\mathcal{S}}_{s_j,s})^2. \tag{4.2.31}$$

Because of (Bias Control), the estimators $\frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})$ and $\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}}$ have negligible bias. Starting from $\delta(s_k) = \overline{s_k}$, we can reasonably assume inductively that (4.2.31) is true for $s = \delta(s_k)$. Afterwards, we choose the appropriate value of $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$, subsequently the value of $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$, that makes $\sigma^{m_0,\mathcal{S}}_{s_j,s_k}$ satisfy (4.2.31) for $s = s_k$. For this task, we introduce an intermediary non-adjusted conditional variance $(\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s})^2$ defined by

$$(\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s_k})^2 = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k} - \sum_{m_1=1}^{M_1}\frac{\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})}{M_1}\right]^2.$$

$\sum_{m_1=1}^{M_1}\frac{\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})}{M_1}$ can be replaced by $\sum_{m_1=1}^{M_1}\frac{\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k}}{M_1}$ without changing the value of $(\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s})^2$. For positive tuning value $\epsilon_3 < 1/3$, we set then

$$\gamma^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}}{\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s_k}}\left(\sqrt{\frac{s_k - s_j}{\delta(s_k) - s_j}}1_{\delta(s_k)-s_j<\epsilon_3} + \frac{\sigma^{\mathcal{S}}_{0,s_j,s_k}}{\sigma^{\mathcal{S}}_{0,s_j,\delta(s_k)}}1_{\delta(s_k)-s_j\geq\epsilon_3}\right). \tag{4.2.32}$$

According to (4.2.32), when $\delta(s_k) - s_j$ is small and a fortiori $s_k - s_j$ is small then the conditional variance $(\sigma^{m_0,\mathcal{S}}_{s_j,s_k})^2$ is linear with respect to time increment $s_k - s_j$. This fact can be justified for diffusions using first order Taylor expansion of $E(\phi(t, W_t))$ around $\phi(t, 0)$, where W is a Brownian motion. Also according to (4.2.32), when $\delta(s_k) - s_j$ becomes sufficiently big, the conditional variance $(\sigma^{m_0,\mathcal{S}}_{s_j,s_k})^2$ has the same unconditional decreasing ratio $\left(\frac{\sigma^{\mathcal{S}}_{0,s_j,s_k}}{\sigma^{\mathcal{S}}_{0,s_j,\delta(s_k)}}\right)^2$ with respect to $(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2$. Although this adjustment works well in our simulations, it can be turned into a more local approach. In fact, similar to what was proposed for the bias control in (4.2.23), for $s_j < s$ with $s = s_k, \delta(s_k)$, the equality

$$\mathbb{E}\Big(\mathrm{Var}_{s_j}(U_s)1_{\{\mathrm{Var}_{s_j}(U_{\delta(s_k)})\in[a,b]\}}\Big) = \mathbb{E}\Big(1_{\{\mathrm{Var}_{s_j}(U_{\delta(s_k)})\in[a,b]\}}\big[U_s - \mathbb{E}_{s_j}(U_s)\big]^2\Big)$$

is true for any localizing interval $[a, b]$. When M_0 is sufficiently large, one can sort $\{(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2\}_{m_0\leq M_0}$ and define a subdivision of localizing intervals $\{[a_q, a_{q+1}]\}_{q\geq1}$ and define

$$(\sigma^{\mathcal{S},q}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}1_{\{(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2\in[a_q,a_{q+1}]\}}\left[\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}} - \frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})\right]^2.$$

Condition (4.2.31) can be then replaced by its localized version

$$(\sigma^{\mathcal{S},q}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}1_{\{(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2\in[a_q,a_{q+1}]\}}(\sigma^{m_0,\mathcal{S}}_{s_j,s})^2. \tag{4.2.33}$$

If $\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)} \in [a_{q_0}, a_{q_0+1}]$ then it makes sense to replace (4.2.32) by

$$\gamma^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}}{\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s_k}}\left(\sqrt{\frac{s_k - s_j}{\delta(s_k) - s_j}}1_{\delta(s_k)-s_j<\epsilon_3} + \frac{\sigma^{\mathcal{S},q_0}_{0,s_j,s_k}}{\sigma^{\mathcal{S},q_0}_{0,s_j,\delta(s_k)}}1_{\delta(s_k)-s_j\geq\epsilon_3}\right) \tag{4.2.34}$$

Although the local variance adjustment (4.2.34) was not necessary in our simulations, it is quite remarkable to point out the high flexibility of the multilayer setting induced by 1NMC. Thus when M_0 and M_1 are sufficiently large, one sees that this double layer Monte Carlo makes possible a very fine tracking of the bias of the first layer fine estimator U^{m_0} and the variance of the second layer coarse estimator U^{m_0,m_1}.

4.3 Some applications : Risk measures, BSDEs and RBSDEs

The simulation procedure presented in the previous section is supposed to be used for any functional approximated by or solution of (f). In this section, we show the use of this procedure on standard problems that inspired this work. We first clarify the method on the approximation of a conditional expectation of some $\mathcal{F}_T$-measurable random variable and how to compute a risk measure. We also illustrate the adaptation to BSDEs then to RBSDEs.

4.3.1 Conditional expectation and risk measures

We consider here the following process

$$U_t = E\left(f(X_T)\Big|X_t\right),$$

with a deterministic function f. Thus, we assume that there is no path dependence through the sum on the realizations of X as done in (f). In this path-independent situation for the fixed time set (4.2.3), it is clear that one can simulate $\{U_{T-s_j^0}\}_{j=0,...,2^L}$ using 1NMC without any need of regression and thus without using our method. However, we choose to illustrate our method on this simple case and we will see at the end of this section what are the benefits. To simplify further the presentation, we set the variance adjustment parameter γ, introduced in Section 4.2.3, to 1.

For known values $s_{j'} < s_j \in \{\Delta_t, ..., T\}$ and for a fixed outer trajectory $(X^{m_0}_{t_k})_{k=0,...,2^L}$, let us assume that we want to simulate $U_{s_{j'}}$ and U_{s_j}. A straight way to do it is to draw inner trajectories of X, as in Figure 4.4, then average on the realizations of $f(X_T)$. If s_j and $s_{j'}$ are close to each other in some sense[1], we are able to simulate U_{t_k} for any $t_k \in [s_{j'}, s_j)$ using

$$U^{m_0}_{t_k} = \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{t_k,T} = \frac{1}{M_1}\sum_{m_1=1}^{M_1} \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j}(X^{m_0,m_1}_{t_k,s_j}).$$

We point out that $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j}(x)$ and $X^{m_0,m_1}_{t_k,s_j}$ replace respectively $f(x)$ and $X^{m_0,m_1}_{t_k,T}$ involved in standard Nested simulation. Below, we establish how $\overline{h}_{s_{j'},s_j}$ should be computed.

First of all, since $s_{j'}$ and s_j are assumed to be "close enough", the initialization phase presented in the end of Section 4.2 and the actualization of $\overline{s_{j'}}$, $\underline{s_{j'}}$, $\overline{s_j}$ and $\underline{s_j}$ are not necessary. Thus, in the light of (4.2.11), one has to take $\overline{s_j} = \overline{s_{j'}} = t_{2^L} = T$ and consequently

$$(\overline{h}_{s_{j'}T})\ \&\ (\overline{h}_{s_jT}) \quad \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T}(x) = \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}(x) = f(x)$$

that sets

$$(\widetilde{h}_{s_j}) \quad U^{m_0}_{s_j} = \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} = \frac{1}{M_1}\sum_{m_1=1}^{M_1} \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}(X^{m_0,m_1}_{s_j,T}).$$

[1] Not necessary an Euclidean distance.

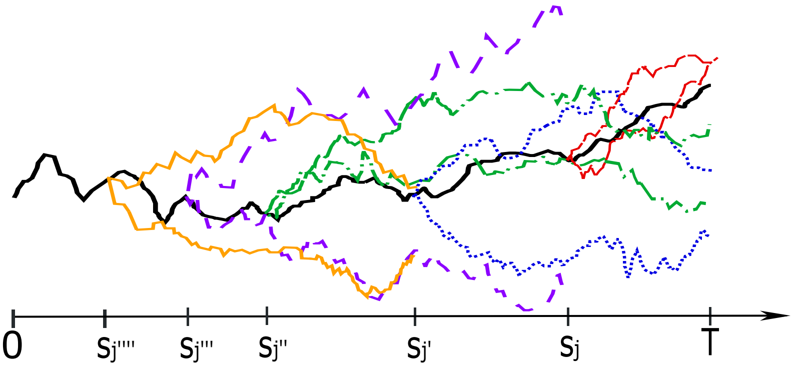


FIGURE 4.4 – Given the realization of one outer trajectory (bold), we simulate inner trajectories to approximate U_{s_j}, $U_{s_{j'}}$, $U_{s_{j''}}$, $U_{s_{j'''}}$ and $U_{s_{j''''}}$.

As in (4.2.15), we define

$$(\overline{h}_{s_{j'}s_j}) \quad \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j}(x) = \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} + {}^t\mathcal{T}^{m_0}_{s_{j'},s_j,M_1'}(x) A^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j},$$

where the adaptation of (4.2.27) and (4.2.30) makes

$$(A^T_{s_{j'}s_j}) \; A^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j} = \frac{\left(\widetilde{\Lambda}^{m_0}_{s_{j'},s_j,M_1'}\right)^{-1}}{M_1} \sum_{m_1=1}^{M_1} \mathcal{T}^{m_0}_{s_{j'},s_j,M_1'}(X^{m_0,m_1}_{s_{j'},s_j}) \begin{bmatrix} \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T}\left(X^{m_0,m_1}_{s_{j'},T}\right) \\ -\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} \end{bmatrix}.$$

If we add a third increment $s_{j''}$ (cf. Figure 4.4) such that $s_{j''}$, $s_{j'}$ and s_j are close enough, for any $t_k \in [s_{j''}, s_{j'})$ one can set $U^{m_0}_{t_k} = \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{t_k,T}$. The latter equality requires the definition of $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}$ which can be obtained from $(\overline{h}_{s_{j''}s_{j'}})$ involving $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T}$ and $A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}$ that can be computed using $(A^{s_j}_{s_{j''}s_{j'}})$. The calculations in $(A^{s_j}_{s_{j''}s_{j'}})$ use $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T}$ and $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}$ whose expression depends on $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}$ and $A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}$. Finally, $A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}$ is the regression vector of $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T}$ around $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}$. Subsequently, the computations of $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T}$, $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T}$ and $\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}$ involve the dependence structure given in (4.3.1).

$$\begin{tikzcd}
\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} & = f \\
 & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j} \arrow[u] \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T} & = f \\
 & & & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} & = f \\
 & & & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} \arrow[u] \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T} & = f
\end{tikzcd} \qquad (4.3.1)$$

By adding other increments $s_{j'''}$ and $s_{j''''}$ (cf. Figure 4.4), it can happen that s_j can no longer be considered close enough. In this situation, a linear regression around $X^{m_0}_{s_j}$ would not be considered sufficient for inner trajectories that start at $X^{m_0}_{s_{j'''}}$ or

$X^{m_0}_{s_{j''''}}$. To deal with this situation, one should introduce $(\overline{s_{j'''}}, \underline{s_{j'''}})$ and $(\overline{s_{j''''}}, \underline{s_{j''''}})$ defined in the end of Section 4.2. For instance if $\overline{s_{j'''}} = s_j$ and $\underline{s_{j'''}} = s_{j''}$, one starts the backward induction associated to the increment $s_{j'''}$ by the final condition $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_j}(x) = \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}(x)$ instead of $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},T}(x) = f(x)$ and (4.3.1) becomes

$$\begin{tikzcd}
\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_j} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j''}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} \dots f \\
 & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j''}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j} \dots f \arrow[u] \\
 & & & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j'}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T} \dots f \\
 & & & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j'}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} \dots f \arrow[u] \\
 & & & & & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_j} = \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} &
\end{tikzcd} \tag{4.3.2}$$

In Figure 4.4, we also set $\overline{s_{j''''}} = s_{j'}$ as well as $\underline{s_{j''''}} = s_{j'''}$ and the tree (4.3.2) can be further changed to include the dependency structure induced by $s_{j''''}$. Indeed, we urge the reader to check that (4.3.2) can be as easily completed as done for (4.3.1) to include the dependency structure induced by $s_{j''''}$.

Even with the simple example presented in this subsection, one can show the benefit of this method. Indeed, in addition to a fine simulation of U using $\widetilde{h}$, this method defines a set of functions $\overline{h}$ that can be considered as coarse conditional approximation of U. These conditional approximations can be used as forward components of another functional. For instance, given the example presented above and illustrated in Figure 4.4, the simulation of an m_0 realization of $V_{s_{j''}} = \mathbb{E}\left((U_{s_j} - U_{s_{j'}})_+ \middle| X_{s_{j''}}\right)$ can be done with

$$\widetilde{V}^{m_0}_{s_{j''}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(\left[\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}(X^{m_0,m_1}_{s_{j''},s_j}) - \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}(X^{m_0,m_1}_{s_{j''},s_{j'}})\right]_+\right).$$

These functions $\overline{h}$ can be also used for risk measures. For example, the conditional value at risk $\mathbb{V}a\mathbb{R}^{\alpha\%}\left[U_{s_j} - U_{s_{j'}} \middle| X_{s_{j''}}\right]$ of level $\alpha\%$ can be computed after sorting $\left(\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}(X^{m_0,m_1}_{s_{j''},s_j}) - \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}(X^{m_0,m_1}_{s_{j''},s_{j'}})\right)_{1\leq m_1\leq M_1}$.

Remark 4.3.1 *Referring to Figure 4.4, for any g, when $\mathbb{E}\left(g(U_{s_{j'''}})\middle|X_{s_{j''''}}\right)$, $\mathbb{E}\left(g(U_{s_{j''}})\middle|X_{s_{j''''}}\right)$ and $\mathbb{E}\left(g(U_{s_{j'}})\middle|X_{s_{j''''}}\right)$ are well defined their simulation can be directly performed using $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_{j'''}}$, $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_{j''}}$ or $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_{j'}}$. This is not the case for $\mathbb{E}\left(g(U_{s_j})\middle|X_{s_{j''''}}\right)$ since $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_j}$ were not computed because $\overline{s_{j''''}} = s_{j'} < s_j$. If $\mathbb{E}\left(g(U_{s_j})\middle|X_{s_{j''''}}\right)$ is needed, one should be either less conservative for the choice of $\epsilon_{1,s_{j''''}}$ and $\epsilon_{2,s_{j''''}}$ (cf. (4.2.11) and (Bias Control)) that makes, or use higher order terms for the regression as presented in Remark 4.2.2.*

The other benefit of our method is the possibility to have a parareal alike implementation [89] and thus make the algorithm parallel in time in addition to have it parallel in paths. Indeed, refering to Figure 4.4, if we associate the final conditions $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}$ and $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j}$ respectively to each subinterval $[s_{j''}, s_{j'})$ and $[s_{j'}, s_j)$, we can perform concurrent calculations on these intervals.

4.3.2 BSDEs with a Markov forward process

In the previous subsection 4.3.1, we saw the implementation of our method on a simple problem and we showed its benefits when one has to simulate functionals of functionals of a Markov process. BSDEs and RBSDEs are specific functionals of functionals of a forward process assumed Markov in various situations. After [100], BSDEs became very widely studied, especially in the quantitative finance community starting with [81]. Here we consider the One step forward Dynamic Programming (ODP) scheme for discrete BSDEs

$$(ODP)\quad Y_T = \zeta \text{ and for } k < 2^L \left\{ \begin{array}{l} Y_{t_k} = \mathbb{E}_{t_k}[Y_{t_{k+1}} + \Delta_t f(t_k, Y_{t_{k+1}}, Z_{t_k})], \\ \quad Z_{t_k} = \mathbb{E}_{t_k}[Y_{t_{k+1}}(W_{t_{k+1}} - W_{t_k})/\Delta_t]. \end{array} \right.$$

(ODP) was studied for instance in [64, 87]. Here we consider $\zeta = f(t_{2^L}, X_T)$ to be some square integrable random variable that depends on X_T. Given a discretization sequence $(s_j)_{j=0,...,2^L} \in \mathcal{S}$ and referring to (4.2.1) and (4.2.2), the simulation of X involves the increments of an $\mathbb{R}^{d_2}$-Brownian motion W with $\xi^{m_0}_{t_k} = W^{m_0}_{t_k} - W^{m_0}_{t_{k-1}}$ and $\xi^{m_0,m_1}_{s_j,t_k} = W^{m_0,m_1}_{s_j,t_k} - W^{m_0,m_1}_{s_j,t_{k-1}}$ where $W^1, ..., W^{M_0}$ are independent realizations of W with

$$W^{m_0,m_1}_{s_j,t_k} = W^{m_0,m_1}_{s_j,t_{k-1}} + \Delta W^{m_0,m_1}_{s_j,t_k} \text{ and } \quad W^{m_0,m_1}_{s_j,s_j}\Big|_{m_1=1,...,M_1+M_1'} = W^{m_0}_{s_j},$$

$(\Delta W^{m_0,m_1}_{s_j,t_k})^{(m_0,m_1)\in\{1,...,M_0\}\times\{1,...,M_1+M_1'\}}_{k\in\{j,...,2^L\},j\in\{1,...,2^L\}}$ are independent Brownian motion increments independent from $W^1, ..., W^{M_0}$ with $\mathbb{E}([\Delta W^{m_0,m_1}_{s_j,t_k}]^2) = \Delta_t$. As pointed out below Remark 4.2.1, if an inner trajectory $\{X^{m_0,m_1}\}$ is needed several times in the backward induction, we simulate independent copies of it and thus independent copies of ξ^{m_0,m_1} and use each copy once.

For given indices $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j}$ and using $\delta_{s_j}(s_k)$ defined in (4.2.14), we also set $\Delta W^{m_0,m_1}_{s_j,s_k,\delta_{s_j}(s_k)} = W^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)} - W^{m_0,m_1}_{s_j,s_k}$. For each k, the Borel $\mathcal{B}(\mathbb{R}) \otimes \mathcal{B}(\mathbb{R}^{d_2})$-measurable driver $f(t_k, \cdot, \cdot)$ is assumed to satisfy Lipschitz condition of Section 4.4.

Given the discretization set $\mathcal{S}$, one can define two coarse approximations around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ given by

$$\overline{y}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \ell\Big[\overline{y}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) C^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{4.3.3}$$

$${}^t\overline{z}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = {}^t\widetilde{z}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) D^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{4.3.4}$$

as well as two fine approximations at $X^{m_0}_s$, for $s \in \{s_k, s_k + \Delta_t, ..., \delta_{s_j}(s_k) - \Delta_t\}$ with $\Delta_s = \delta_{s_j}(s_k) - s$ and $\Delta_{s_k} = \delta_{s_j}(s_k) - s_k$, given by

$$\widetilde{y}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1} \sum_{m_1=1}^{M_1} \left[\begin{array}{c} \Delta_s f(s_k, \overline{y}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)}), \widetilde{z}^{m_0,\mathcal{S}}_{s,\overline{s_k}}) \\ + \overline{y}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)}) \end{array} \right], \tag{4.3.5}$$

$$\widetilde{z}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1\Delta_s}\sum_{m_1=1}^{M_1} \overline{y}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)})\left(W^{m_0,m_1}_{s,\delta_{s_j}(s_k)} - W^{m_0}_{s}\right) \tag{4.3.6}$$

and we set the final coarse approximation to

$$\overline{y}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}} = \begin{cases} f(t_{2^L}, X^{m_0,m_1}_{s_j,t_{2^L}}) & \text{if } \overline{s_j} = t_{2^L}, \\ \overline{y}^{m_0,\mathcal{S}}_{\underline{s_j},\overline{s_j}}(X^{m_0,m_1}_{s_j,\overline{s_j}}) = \overline{y}^{m_0,\mathcal{S}}_{\delta_{s_j}(s_j),\overline{s_j}}(X^{m_0,m_1}_{s_j,\overline{s_j}}) & \text{if } \overline{s_j} < t_{2^L}, \end{cases} \tag{4.3.7}$$

$\overline{s_j} > \underline{s_j} > s_j$ are specified during the initialization phase (cf. (4.2.12)) then actualized at each step (cf. (4.2.25) and (4.2.26)) where (Bias Control), ε and e are expressed in Definition 4.3.1.

Since $\mathcal{T}$ was already expressed in (4.2.6), to complete this inductive interconnected definition of ($\overline{y}$, $\widetilde{y}$, $\overline{z}$, $\widetilde{z}$), we set the vector $C^{m_0,\mathcal{S}}_{s_j,s_k} = \gamma^{m_0,\mathcal{S}}_{s_j,s_k}\widehat{C}^{m_0,\mathcal{S}}_{s_j,s_k}$ and the matrix $D^{m_0,\mathcal{S}}_{s_j,s_k}$ to be equal to

$$\widehat{C}^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{(\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}}{M_1}\sum_{m_1=1}^{M_1}\mathcal{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}), \tag{4.3.8}$$

$$D^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{(\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}}{M_1}\sum_{m_1=1}^{M_1}\mathcal{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k}, \Delta W^{m_0,m_1}_{s_j,s_k,\delta_{s_j}(s_k)}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}) \tag{4.3.9}$$

with $\mathcal{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x',x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x')\mathbb{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x',x)$ is $\mathcal{F}_{\delta_{s_j}(s_k)} \otimes \mathcal{B}(\mathbb{R}^{2d_1})-$ measurable and $\mathcal{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x',w,x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x')^t\mathbb{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(w,x)$ is a vector function measurable with respect to $\mathcal{F}_{\delta_{s_j}(s_k)} \otimes \mathcal{B}(\mathbb{R}^{2d_1+d_2})$, where

$$\mathbb{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x',x) = \begin{bmatrix} \Delta_{s_k} f(s_k, \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x), \overline{z}^{m_0,\mathcal{S}}_{s_j,s_k}(x')) \\ +\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x) - \widetilde{y}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} \end{bmatrix}, \tag{4.3.10}$$

and

$$\mathbb{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(w,x) = \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x)\frac{w}{\Delta_{s_k}} - \widetilde{z}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}. \tag{4.3.11}$$

Finally, applying similar variance adjustment procedure as the one presented in Section 4.2.3, we set the value of $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$ and we define

$$\ell\Big[\overline{y}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] = \widetilde{y}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + \frac{(1-\gamma^{m_0,\mathcal{S}}_{s_j,s_k})}{M_1}\sum_{m_1=1}^{M_1} {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{C}^{m_0,\mathcal{S}}_{s_j,s_k}. \tag{4.3.12}$$

From equations above, one can associate quadratic minimization problems to $C^{m_0,i}_{s_j,s_k}$ and to $D^{m_0,i}_{s_j,s_k}$, as done for $H^{m_0,i}_{s_j,s_k}$ in (4.2.18). In the same fashion as in Definition 4.2.1, we define the double layer approximations (Y^{m_0}, Z^{m_0}) and $(Y^{m_0,m_1}, Z^{m_0,m_1})$ of functionals Y and Z.

Definition 4.3.1 *For* $i^* = \min(\min\{i = 1, ..., L-L', j_i^* = 2^L\}, L-L')$

- *For $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j} < t_{2^L} = T$, the simulation $Y^{m_0,m_1}_{s_j,s_k}$ and $Z^{m_0,m_1}_{s_j,s_k}$ of Y and Z respectively around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ are set to be equal to $\overline{y}^{m_0,\mathcal{S}i^*}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ and $\overline{z}^{m_0,\mathcal{S}i^*}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ where $\overline{y}$ and $\overline{z}$ are given in* (4.3.3), (4.3.4) *and* (4.3.7).
- *For $k \in \{1, ..., 2^L\}$ and $s \in \{s_k, s_k + \Delta_t, ..., \delta_{s_k}(s_k) - \Delta_t\} - \{0\}$, the simulation $Y^{m_0}_s$ and $Z^{m_0}_s$ of Y and Z respectively at $X^{m_0}_s$ are set to be equal to $\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{s,\overline{s_k}}$ and to $\widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{s,\delta_{s_k}(s_k)}$ with $\widetilde{y}$ and $\widetilde{z}$ expressed in* (4.3.5) *and* (4.3.6).
- *The average Y^{lear}_0 and Z^{lear}_0 of learned values on Y_0 and Z_0 are respectively equal to*

$$Y^{lear}_0 = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}}, \quad Z^{lear}_0 = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}} \tag{4.3.13}$$

and the simulated values Y^{sim}_0 and Z^{sim}_0 of Y_0 and Z_0 are respectively equal to

$$\begin{aligned} Y^{sim}_0 &= \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\delta(0) f\left(\delta(0), \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}, Z^{sim}_0\right) + \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\right], \\ Z^{sim}_0 &= \sum_{m_0=1}^{M_0} \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}} \frac{W^{m_0}_{\delta(0)}}{\delta(0) M_0}. \end{aligned} \tag{4.3.14}$$

- *Introduced in* (4.2.10), *(Bias Control) associated to* (*ODP*) *is defined at $s \in \mathcal{S}^0$ for $u \in \mathcal{S}^0 \cap]s, \overline{\delta(s)}]$ by*

$$\left|\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{y}^{m_0,\mathcal{S}^0}_{s,u} - \widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}} - (\delta(s) - s) f(s, \widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}}, \widetilde{z}^{m_0,\mathcal{S}^0}_{s,\overline{s}})\right)\right| < \epsilon^{\mathcal{S}^0}_{2,s}$$

where for each set $\mathcal{S}$, $\{\epsilon^{\mathcal{S}}_{2,s}\}_{s\in\mathcal{S}}$ is a family of positive bias tuning parameters.
- *For $k \in \{j^*_i + 1, ..., 2^L\}$, setting $s_k = T - s^i_k$ and noticing that $\delta^{\mathcal{S}^i}(s_k) = \delta^{\widehat{\mathcal{S}}^i}(\delta^{\widehat{\mathcal{S}}^i}(s_k))$, $e^{\mathcal{S}^i}_{s_k}$ and $\varepsilon^{\mathcal{S}^i}_{s_k}$ are given by*

$$e^{\mathcal{S}^i}_{s_k} = \frac{1}{M_0 M_1}\sum_{m_0=1}^{M_0}\sum_{m_1=1}^{M_1}\left[\overline{y}^{m_0,\widehat{\mathcal{S}}^i}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}) - \overline{y}^{m_0,\mathcal{S}^i}_{s_k,\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)})\right],$$

$$\varepsilon^{\mathcal{S}^i}_{s_k} = \sum_{s\in\mathcal{S}^i, s>s_k} \epsilon^{\mathcal{S}^i}_{2,s}.$$

Remark 4.3.2 *0. The different points of Remark 4.2.3 can be highlighted here.*

1. *Given a discretization set $\mathcal{S}$ and $s_k \in \mathcal{S}$, the choice of $\underline{s_k}$ and on $\overline{s_k}$ is completely known in Definition 4.3.1 through the value of e, ε and inequality (Bias Control).*
2. *The value of e, ε and inequality (Bias Control) involve mainly the approximation of Y since using criteria on the approximation of Z would involve very large number of trajectories, making it impracticable.*
3. *Although possible, we did not judge necessary to implement a variance adjustment method on the Z component.*

4. *As a future work, we would like to apply variance reduction methods with 1NMC and provide very accurate double layer estimations of the Z term.*
5. *With BSDEs, it is possible to use other (Bias Control) inequalities. Indeed, using rather an MDP scheme (cf. [67]), (Bias Control) of Definition 4.3.1 can be replaced by*

$$\left|\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{y}^{m_0,\mathcal{S}^0}_{s,u}-\widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(r),\overline{\delta(r)}}-\sum_{\theta\in\mathcal{S}^0,\theta=s}^{r}(\delta(\theta)-\theta)f(s,\widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(\theta),\overline{\delta(\theta)}},\widetilde{z}^{m_0,\mathcal{S}^0}_{\theta,\overline{\theta}})\right)\right|<\epsilon^{\mathcal{S}^0}_{2,s},$$

for any $r\in\mathcal{S}^0\cap[s,u[$.

4.3.3 RBSDEs with a Markov forward process

The generally studied RBSDEs are functionals of a Markov process. Here, we consider an application to RBSDEs as the one presented in [35] with X simulated like in Section 4.3.3 and functions $g(\cdot)$ and driver $\{f(t_k,\cdot)\}_{k=0}^{2^L-1}$ assumed to satisfy Lipschitz condition of Section 4.4. We want to propose a double layer approximation V^{m_0} and V^{m_0,m_1} of the Snell envelope V, solution to

$$(Snl)\ V_T=g(X_T)\text{ and for }k<2^L:V_{t_k}=g(X_{t_k})\vee\mathbb{E}_{t_k}[V_{t_{k+1}}+\Delta_t f(t_k,V_{t_{k+1}})],$$

that can be done using straightforwardly the recipe of Section 4.2 combined with a maximization by g. In fact, given a discretization set $\mathcal{S}$ and indices $k<j\in\{1,...,2^L\}$ that satisfy $s_j<s_k\leq\overline{s_j}$ and using $\delta_{s_j}(s_k)$ defined in (4.2.14), we set the coarse approximation $\overline{v}_{s_j,s_k}$ around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ to

$$\overline{v}^{m_0,\mathcal{S}}_{s_j,s_k}(x)=\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}(x)\vee g(x),\tag{4.3.15}$$

and the fine approximation $\widetilde{v}_{s,\overline{s_k}}$ at $X^{m_0}_s$, $s\in\{s_k,s_k+\Delta_t,...,\delta_{s_j}(s_k)-\Delta_t\}$, to

$$\widetilde{v}^{m_0,\mathcal{S}}_{s,\overline{s_k}}=\widetilde{w}^{m_0,\mathcal{S}}_{s,\overline{s_k}}\vee g(X^{m_0}_s).\tag{4.3.16}$$

Denoting $\Delta_s=\delta_{s_j}(s_k)-s$ and $\Delta_{s_k}=\delta_{s_j}(s_k)-s_k$, we define

$$\widetilde{w}^{m_0,\mathcal{S}}_{s,\overline{s_k}}=\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(\Delta_s f(s_k,\overline{v}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)}))+\overline{v}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)})\right),\tag{4.3.17}$$

$$\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}(x)=\ell\left[\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}\right]+{}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x)B^{m_0,\mathcal{S}}_{s_j,s_k},\tag{4.3.18}$$

where

$$\ell\left[\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}\right]=\widetilde{w}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}+\frac{(1-\gamma^{m_0,\mathcal{S}}_{s_j,s_k})}{M_1}\sum_{m_1=1}^{M_1}{}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{B}^{m_0,\mathcal{S}}_{s_j,s_k},\tag{4.3.19}$$

$B^{m_0,\mathcal{S}}_{s_j,s_k}=\gamma^{m_0,\mathcal{S}}_{s_j,s_k}\widehat{B}^{m_0,\mathcal{S}}_{s_j,s_k}$ with

$$\widehat{B}^{m_0,\mathcal{S}}_{s_j,s_k}=\frac{(\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}}{M_1}\sum_{m_1=1}^{M_1}\mathcal{B}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k},X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)})$$

and $\mathcal{B}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x',x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x')\mathbb{B}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x)$ with

$$\mathbb{B}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x) = \left[\begin{array}{c} \Delta_{s_k} f(s_k, \overline{v}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x)) \\ +\,\overline{v}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x) - \widetilde{w}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} \end{array}\right], \tag{4.3.20}$$

with a final coarse approximation given by

$$\overline{v}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}(x) = \begin{cases} g(x) & \text{if } \overline{s_j} = t_{2^L}, \\ \overline{v}^{m_0,\mathcal{S}}_{\underline{s_j},\overline{s_j}}(x) = \overline{v}^{m_0,\mathcal{S}}_{\delta_{s_j}(s_j),\overline{s_j}}(x) & \text{if } \overline{s_j} < t_{2^L}, \end{cases} \tag{4.3.21}$$

where $\overline{s_j} > \underline{s_j} > s_j$ are specified during the initialization phase (cf. (4.2.12)) then actualized at each step (cf. (4.2.25) and (4.2.26)).

Definition 4.3.2 *For* $i^* = \min(\min\{i = 1, ..., L - L', j_i^* = 2^L\}, L - L')$

— *For* $k < j \in \{1, ..., 2^L\}$ *that satisfy* $s_j < s_k \leq \overline{s_j} < t_{2^L} = T$, *the simulation* $V^{m_0,m_1}_{s_j,s_k}$ *of* V *around* $X^{m_0}_{s_k}$ *conditionally on* $X^{m_0}_{s_j}$ *is set to be equal to* $\overline{v}^{m_0,\mathcal{S}^{i^*}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ *where* $\overline{v}$ *is given in* (4.3.15), (4.3.18) *and* (4.3.21).

— *For* $k \in \{1, ..., 2^L\}$ *and* $s \in \{s_k, s_k + \Delta_t, ..., \delta_{s_k}(s_k) - \Delta_t\} - \{0\}$, *the simulation* $V^{m_0}_s$ *of* V *at* $X^{m_0}_s$ *is set to be equal to* $\widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{s,\overline{s_k}}$ *with* $\widetilde{v}$ *expressed in* (4.3.16) *and* (4.3.17).

— *The average* V_0^{lear} *of the learned values on* V_0 *is equal to*

$$V_0^{lear} = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}}, \tag{4.3.22}$$

and the simulated values V_0^{sim} *of* V_0 *is equal to*

$$V_0^{sim} = g(x_0) \vee \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\delta(0) f\left(\delta(0), \widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\right) + \widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\right]. \tag{4.3.23}$$

— *Introduced in* (4.2.10), *(Bias Control) associated to (Snl) is defined at* $s \in \mathcal{S}^0$ *for* $u \in \mathcal{S}^0 \cap]s, \overline{\delta(s)}]$ *by*

$$\left|\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{w}^{m_0,\mathcal{S}^0}_{s,u} - \widetilde{v}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}} - (\delta(s) - s) f(s, \widetilde{v}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}})\right)\right| < \epsilon^{\mathcal{S}^0}_{2,s}$$

where for each set $\mathcal{S}$, $\{\epsilon^{\mathcal{S}}_{2,s}\}_{s\in\mathcal{S}}$ *is a family of positive bias tuning parameters.*

— *For* $k \in \{j_i^* + 1, ..., 2^L\}$, *setting* $s_k = T - s_k^i$ *and noticing that* $\delta^{\mathcal{S}^i}(s_k) = \delta^{\widehat{\mathcal{S}}^i}(\delta^{\widehat{\mathcal{S}}^i}(s_k))$, $e^{\mathcal{S}^i}_{s_k}$ *and* $\varepsilon^{\mathcal{S}^i}_{s_k}$ *are given by*

$$e^{\mathcal{S}^i}_{s_k} = \frac{1}{M_0 M_1}\sum_{m_0=1}^{M_0}\sum_{m_1=1}^{M_1}\left[\overline{w}^{m_0,\widehat{\mathcal{S}}^i}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}) - \overline{w}^{m_0,\mathcal{S}^i}_{s_k,\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)})\right],$$

$$\varepsilon^{\mathcal{S}^i}_{s_k} = \sum_{s\in\mathcal{S}^i, s>s_k} \epsilon^{\mathcal{S}^i}_{2,s}.$$

Remark 4.3.3 *0. The different points of Remark 4.2.3 can be highlighted here.*

1. *Given a discretization set $\mathcal{S}$ and $s_k \in \mathcal{S}$, the choice of $\underline{s_k}$ and on $\overline{s_k}$ is completely known in Definition 4.3.2 through the value of e, ε and inequality (Bias Control).*
2. *Unlike BSDEs, it is not possible to use an MDP scheme for (Bias Control) as explained in Remark 4.3.2.5.*
3. *Although using an optimal stopping formulation [42] of the dynamic programming is known to provide better numerical results [90], we prefared here to use NMC on the top of the original algorithm [113] since its error estimates remains similar to the one presented in Section 4.4 for BSDEs.*
4. *As a future work, we would like to apply variance reduction methods with 1NMC and provide very accurate double layer estimations of the optimal stopping strategy.*

4.4 Error estimates and cutting bias propagation

After expressing error estimates for both coarse and fine approximations in Section 4.4.1, we show how to cut bias propagation using our new judicious trick presented in Section 4.4.2.

4.4.1 Regression-based NMC and increasing the learning depth

Before presenting the main elements, we point out that we have intentionally considered only discrete functionals of a Markov process. The approximation due to discretization of the continuous version of BSDEs is not studied and we refer to [64, 87] among others that quantify well the resulting error. Moreover, we also consider the discretized version of the Markov process introduced in (4.2.1) and (4.2.2) where

$$\mathcal{E}_{t_k}(x, \xi) = x + \Delta_t b(t_k, x) + \sigma(t_k, x)\xi \tag{4.4.1}$$

with the usual (cf. [98]) Lipschitz continuity condition on the coefficients $b(t, x)$ and $\sigma(t, x)$ uniformly with respect to $t \in [0, T]$. Similar to what was considered in Section 4.3.2, the noise ξ is given by increments of a vector of independent Brownian motions i.e. $\xi^{m_0}_{t_k} = W^{m_0}_{t_k} - W^{m_0}_{t_{k-1}}$ and $\xi^{m_0,m_1}_{s_j,t_k} = W^{m_0,m_1}_{s_j,t_k} - W^{m_0,m_1}_{s_j,t_{k-1}}$. (4.4.1) can be read as an Euler scheme of a stochastic differential equation that admits a strong solution. In this paper, when the discretization is needed, we assume that L is sufficiently large to neglect the discretization error of the forward process X.

Given two arbitrary square integrable random variables χ_1 and χ_2, consider $\{\overline{\chi_3}^{m_1}\}_{m_1=1}^{M_1}$ to be the empirical regression of χ_1 with respect to χ_2, the authors of [37] established an upper bound error of the regression-based NMC estimator $\dfrac{1}{M_1}\sum\limits_{m_1=1}^{M_1} \phi(\overline{\chi_3}^{m_1})$ of $\mathbb{E}(\phi(\mathbb{E}(\chi_1|\chi_2)))$ once we know the representation error $\kappa = \mathbb{E}(\chi_1|\chi_2) - {}^t R\mathcal{B}(\chi_2)$ induced by the projection of $\mathbb{E}(\chi_1|\chi_2)$ on the basis $\mathcal{B}(\chi_2)$. The fine approximations $\widetilde{h}$, $\widetilde{y}$ and $\widetilde{w}$ presented earlier were computed by averaging on the empirical regressions $\overline{h}$, $\overline{y}$ and $\overline{v}$. It is then interesting to see how to control the error of the fine approximations through the representation error like in [37].

First, for $s_j < s_k < \overline{s_j}$ and Borel measurable Θ function of $(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})$ with $\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})$ integrable, we denote $\mathbb{E}^{m_0,x}_{s_j,s_k}$, $\overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}$ and $\widehat{\mathbb{E}}^{m_0,x}_{s_j,s_k}$ the operators defined by

$$\mathbb{E}^{m_0,x}_{s_j,s_k}(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})) = \mathbb{E}^{m_0}_{s_j}\left(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})|X^{m_0,m_1}_{s_j,s_k} = x\right),$$

$$\begin{aligned}&\overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)}))\\ &= {}^t\mathcal{T}^{m_0}_{s_j,s_k}(x - X^{m_0}_{s_k})\frac{\left(\overline{\Lambda}^{m_0}_{s_j,s_k}\right)^{-1}}{M_1}\sum_{m_1=1}^{M_1}\left[\mathcal{T}^{m_0}_{s_j,s_k}\left(X^{m_0,m_1}_{s_j,s_k} - X^{m_0}_{s_k}\right)\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})\right]\end{aligned}$$

and

$$\widehat{\mathbb{E}}^{m_0,x}_{s_j,s_k}(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})) = {}^t\mathcal{T}^{m_0}_{s_j,s_k}(x - X^{m_0}_{s_k})R^{m_0}_{s_j,s_k}\left[\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})\right]$$

with

$$R^{m_0}_{s_j,s_k}\left[\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})\right] \in \underset{r\in\mathbb{R}^{d_1}}{\operatorname{argmin}}\,\mathbb{E}^{m_0}_{s_j}\left(\begin{bmatrix}\mathbb{E}^{m_0}_{s_j}\left(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})|X^{m_0,m_1}_{s_j,s_k}\right)\\ -{}^t\mathcal{T}^{m_0}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k} - X^{m_0}_{s_k})r\end{bmatrix}^2\right)$$

When $\mathbb{E}_{s_j}$ is the conditional expectation knowing $X^{m_0}_{s_j}$, $\mathbb{E}^{m_0}_{s_j}$ is the conditional expectation knowing the trajectory of X^{m_0} starting from $X^{m_0}_{s_j}$. $\mathbb{E}^{m_0}_{s_j}$ is used as the regression basis depends on X^{m_0}. For a given $s_j \in \mathcal{S}$, in contrast to expressions presented in sections 4.2.2, 4.3.2 and 4.3.3, we simplify the presentation here and we omit to center the regressions around $\widetilde{h}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{\delta(s_j)}}$, $\widetilde{y}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{\delta(s_j)}}$ or $\widetilde{w}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{\delta(s_j)}}$. Consequently, the value of $\widetilde{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$, $\widetilde{y}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$ and $\widetilde{w}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$ are obtained through respectively averaging on $\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}(\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}))$, $\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}(\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}))$ and on $\overline{v}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(x) = g(x) \vee \overline{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}(\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}))$, where

$$\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}) = \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)}) + \sum_{t_l\geq\delta(s_j)}^{\delta^2(s_j)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}),$$

$$\begin{aligned}\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}) = {}&\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)})\\ &+\Delta_{\delta(s_j)}f(\delta(s_j), \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)}), \overline{z}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(X^{m_0,m_1}_{s_j,\delta(s_j)})),\end{aligned}$$

$$\begin{aligned}\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}) = {}&\overline{v}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)})\\ &+\Delta_{\delta(s_j)}f(\delta(s_j), \overline{v}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)})).\end{aligned}$$

We assume Lipschitz condition uniformly in time of the driver f involved in (ODP) and (Snl) with respect to its Y and Z coordinates or with respect to its V coordinate. Although this conditions are not necessary to obtain good numerical results in Section 4.5, they are required to apply Theorem 2 of [37] (cf. Assumption F2 in [37]) that yield the following asymptotical result.

Proposition 4.4.1 *Given that assumptions A1, A2 and A3 of [37]) are fulfilled and that both drivers involved in (ODP) and in (Snl) are $[f]_{Lip}$-Lipschitz we have the following asymptotical inequality*

$$(\widetilde{\rho} - \rho)^2 \leq [\rho]_{Lip} \mathbb{E}^{m_0}_{s_j}(\kappa^2(X^{m_0,m_1}_{s_j,\delta(s_j)})) + O_p(1/M_1) \tag{4.4.2}$$

as $M_1 \longrightarrow \infty$ where $(\widetilde{\rho}, \rho, [\rho]_{Lip}, \kappa)$ is either equal to $(\widetilde{\rho}^h, \rho^h, [\rho]^h_{Lip}, \kappa^h)$ for (f), $(\widetilde{\rho}^y, \rho^y, [\rho]^y_{Lip}, \kappa^y)$ for (ODP) or equal to $(\widetilde{\rho}^v, \rho^v, [\rho]^v_{Lip}, \kappa^v)$ for (Snl) with $\widetilde{\rho}^h = \widetilde{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$, $\widetilde{\rho}^y = \widetilde{y}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$, $\widetilde{\rho}^v = \widetilde{w}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$,

$$\rho^h = \mathbb{E}^{m_0}_{s_j}\left(\mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] + \sum_{t_l \geq s_j}^{\delta(s_j)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right),$$

$$\rho^y = \mathbb{E}^{m_0}_{s_j}\left(\mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right]\right.$$
$$\left. + \Delta_{s_j} f\left(s_j, \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right], \widetilde{z}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}\right)\right),$$

$$\rho^v = \mathbb{E}^{m_0}_{s_j}\left(g(X^{m_0,m_1}_{s_j,\delta(s_j)}) \vee \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right]\right.$$
$$\left. + \Delta_{s_j} f\left(s_j, g(X^{m_0,m_1}_{s_j,\delta(s_j)}) \vee \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right]\right)\right),$$

$[\rho]^h_{Lip} = 1$, $[\rho]^y_{Lip} = 1 + \Delta_{s_j}[f]_{Lip}$, $[\rho]^v_{Lip} = 1 + \Delta_{s_j}[f]_{Lip}$ and

$$\kappa^h(x) = \mathbb{E}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] - \widehat{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right],$$

$$\kappa^y(x) = \mathbb{E}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] - \widehat{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right],$$

$$\kappa^v(x) = \mathbb{E}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] - \widehat{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right].$$

Proposition 4.4.1 results from Theorem 2 and Remark 2 of [37] ; we expressed $[\rho]_{Lip}$ associated to each problem and we replaced $\mathbb{E}$ by $\mathbb{E}^{m_0}_{s_j}$ as the regression basis depends on X^{m_0}. Assumptions A1, A2 and A3 of [37] are standard assumptions for regressions (cf. [114]). Considering the regression basis presented in Section 4.2.1 with $\mathbb{E}(|X_t|^2) < \infty$ for any $t \in [0,T]$, these assumptions are fulfilled if : *i*) the conditional variance of each regressed quantity is integrable and bounded from below by $v_0 > 0$, *ii*) the regression value is unbiased and *iii*) each component of the regression basis as well as κ (denoted M in [37]) admit a finite fourth moment. When the latter moment assumption *iii* is needed to establish error control and can be modified using truncation (cf. [65]), the further *i*&*ii* are sufficient to ensure the existence and uniqueness of the regressed representation.

In Proposition 4.4.1, we provided a control on fine approximations $\widetilde{h}$, $\widetilde{y}$ and $\widetilde{v}$. In Proposition 4.4.2, we rather focus on coarse approximations and decompose the conditional mean square error $\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$ into a bias term $\mathcal{W}$, a variance term $\mathcal{V}$ and a regression error term $\mathcal{R}$.

Proposition 4.4.2 *Assuming i and iii introduced above, for $s_j < s < s_k$ taking their values in the discretization set $\mathcal{S}$, we define*

$$\mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \mathbb{E}^{m_0}_{s_j}\Big(\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x) - \mathcal{U}^{m_0,\mathcal{S}}_{s_j,s_k}(x)\Big),$$

$$\mathcal{R}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \mathbb{E}^{m_0}_{s_j}\Big(\mathcal{U}^{m_0,\mathcal{S}}_{s_j,s_k}(x) - U_{s_k}(x)\Big),$$

$$\mathcal{V}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \mathit{Var}^{m_0}_{s_j}\Big(\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x)\Big),$$

with

$$\mathcal{U}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}\left(U_{\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) + \sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right)$$

then

$$\begin{aligned}\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big) &= \mathbb{E}^{m_0}_{s_j}(\mathcal{V}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}))\\ &+\mathbb{E}^{m_0}_{s_j}\Big([\mathcal{R}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) + \mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)\end{aligned}$$

and there exists a positive constant $\mathcal{K}^{m_0}_{1,s_j,s_k}$ depending on the regression basis such that

$$\mathbb{E}^{m_0}_{s_j}\Big([\mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big) \leq \mathcal{K}^{m_0}_{1,s_j,s_k}\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) - U_{\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)})]^2\Big).$$

Proof *As we simulate several independent copies of X^{m_0,m_1} (cf. the paragraph under Remark 4.2.1), we make sure that the approximations $\overline{h}$ are independent from X^{m_0,m_1} conditionally on X^{m_0}. Then, the expansion of the conditional mean square error $\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$ can be obtained when we notice that*

$$U_{s_k}(X^{m_0,m_1}_{s_j,s_k}) = \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,s_k}}_{s_j,s_k}\left(U_{\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) + \sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right).$$

An expression for the constant $\mathcal{K}^{m_0}_{1,s_j,s_k}$ can be obtained after expanding $\mathbb{E}^{m_0}_{s_j}\Big([\mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$ using

$$\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}\left(\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) + \sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right).$$

Finally, we should point out that one could establish a similar result for (ODP) and (Snl). Indeed, for instance, using the following coarse discretization to approximate (ODP)

$$\begin{cases}\widehat{Y}_{s_k} &= \quad \mathbb{E}_{s_k}\Big(\widetilde{f}_{s_k}(\widehat{Y}_{\delta(s_k)}, \widehat{Z}_{s_k})\Big),\\ \widehat{Z}_{s_k} &= \quad \frac{1}{\Delta_{s_k}}\mathbb{E}_{s_k}\Big(\widehat{Y}_{\delta(s_k)}(W_{\delta(s_k)} - W_{s_k})\Big),\end{cases} \tag{4.4.3}$$

with $\widetilde{f}_{s_k}(y,z) = y + \Delta_{s_k} f_{s_k}(y,z)$, the bias is then controlled as follows

$$\mathbb{E}^{m_0}_{s_j}\Big([\widetilde{\mathcal{W}}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big) \leq \widetilde{\mathcal{K}}^{m_0}_{1,s_j,s_k}\mathbb{E}^{m_0}_{s_j}\left(\begin{bmatrix} \widetilde{f}_{s_k}(\widehat{Y}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}), \widehat{Z}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})) \\ -\widetilde{f}_{s_k}(\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}), \overline{z}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}))\end{bmatrix}^2\right),$$

for some positive constant $\widetilde{\mathcal{K}}^{m_0}_{1,s_j,s_k}$ depending on the regression basis. Therefore, the bias upper bound depends heavily on the driver choice. In the case of (Snl), g also plays an important role on the nonlinearity and subsequently on bias.

4.4.2 Regression with different starting points

As shown in Proposition 4.4.2, the bias $\mathcal{W}$ at time step s_k is controlled by the mean square error at time step $\delta(s_k)$ decomposed into a variance term $\mathcal{V}$, a regression error term $\mathcal{R}$ and a bias term at time step $\delta(s_k)$. Thus, increasing the number of time steps weaken the bias control as it involves more and more terms. In some situations, this accumulation of errors is a source of a significant bias back propagation. In this paper, we proposed a new approximation trick to cut this bias back propagation.

In this section, we present a control on this new approximation that is used twice in the generic presentation of our method in Section 4.2.2. This same approximation was also adapted in Section 4.3.2 to BSDEs and in Section 4.3.3 to RBSDEs. In the generic situation, equations (4.2.16) defines $\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_k}}$ for any $s \in \{s_k, s_k{+}\Delta_t, ..., \delta_{s_j}(s_k){-}\Delta_t\}$ using $\overline{h}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(\cdot)$ which is deduced from a regression on $X^{m_0,m_1}_{s_k,\delta_{s_j}(s_k)}$ instead of a regression on $X^{m_0,m_1}_{s,\delta_{s_j}(s_k)}$. Said differently, provided that s is sufficiently close to s_k we replaced a regressed function obtained from inner trajectories starting at s by a regressed function obtained from inner trajectories starting at s_k on the same outer trajectory m_0. We did more or less the same thing in (4.2.17) when $\overline{s_j} < T$ as we defined $\overline{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$ to be equal to $\overline{h}^{m_0,\mathcal{S}}_{\delta_{s_j}(s_j),\overline{s_j}}$ i.e. we replaced a regression on $X^{m_0,m_1}_{s_j,\overline{s_j}}$ by a regression on $X^{m_0,m_1}_{\delta_{s_j}(s_j),\overline{s_j}}$. The adaptations of (4.2.16) yield similar approximations in (4.3.5), (4.3.6) and (4.3.17). In the same fashion, the adaptations of (4.2.17) yield similar approximations in (4.3.7) and (4.3.21).

For $s_j < s < s_k$, we summarize both situations saying that the regressed function $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(\cdot)$ resulting form the projection of $\sum\limits_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})$ on $X^{m_0,m_1}_{s_j,s_k}$ is approximated by the regressed function $\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(\cdot)$ resulting form the projection of $\sum\limits_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$ on X^{m_0,m_1}_{s,s_k} and vice versa. This approximation is not absurd since one can straightforwardly see, from the Markov property, that

$$\begin{aligned} U_{s_k}(x) &= \mathbb{E}\left(\sum_{t_l \geq s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\Big| X^{m_0,m_1}_{s_j,s_k} = x\right) \\ &= \mathbb{E}\left(\sum_{t_l \geq s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})\Big| X^{m_0,m_1}_{s,s_k} = x\right). \end{aligned} \tag{4.4.4}$$

To establish a control on $\mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right)$ and on

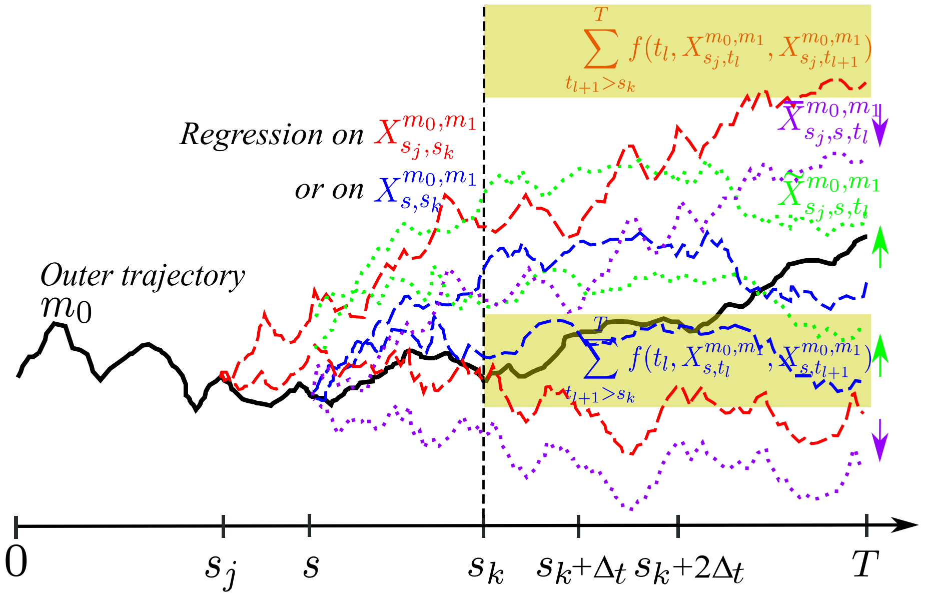

$\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})$
Regression on $X^{m_0,m_1}_{s_j,s_k}$
or on X^{m_0,m_1}_{s,s_k}
Outer trajectory m_0
$\overline{X}^{m_0,m_1}_{s_j,s,t_l}$
$\widetilde{X}^{m_0,m_1}_{s_j,s,t_l}$
$\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$
0
s_j
s
s_k
$s_k{+}\Delta_t$
$s_k{+}2\Delta_t$
T

FIGURE 4.5 – Comparing regression of $\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})$ on $X^{m_0,m_1}_{s_j,s_k}$ and of $\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$ on X^{m_0,m_1}_{s,s_k} with $s \in \{s_j+\Delta_t, ..., s_k-\Delta_t\}$.

$\mathbb{E}^{m_0}_s\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right)$, we define for $t_l \geq s$ two auxilary processes $\overline{X}$ and $\widetilde{X}$ as

$$\begin{cases} \overline{X}^{m_0,m_1}_{s_j,s,s} = X^{m_0}_s, \ \widetilde{X}^{m_0,m_1}_{s_j,s,s} = X^{m_0,m_1}_{s_j,s} \text{ and for } t_l = s+\Delta_t, ..., T \\ \overline{X}^{m_0,m_1}_{s_j,s,t_l} = \mathcal{E}_{t_{l-1}}(\mathcal{E}_{t_{l-2}}(...\mathcal{E}_s(X^{m_0}_s, \xi^{m_0,m_1}_{s_j,s+\Delta_t}), ...\xi^{m_0,m_1}_{s_j,t_{l-1}}), \xi^{m_0,m_1}_{s_j,t_l}) \\ \widetilde{X}^{m_0,m_1}_{s_j,s,t_l} = \mathcal{E}_{t_{l-1}}(\mathcal{E}_{t_{l-2}}(...\mathcal{E}_s(X^{m_0,m_1}_{s_j,s}, \xi^{m_0,m_1}_{s,s+\Delta_t}), ...\xi^{m_0,m_1}_{s,t_{l-1}}), \xi^{m_0,m_1}_{s,t_l}). \end{cases} \tag{4.4.5}$$

where $\mathcal{E}$ is given in (4.4.1). We remind that $E^{m_0}_{s_j}$ and $\mathbb{E}^{m_0}_s$ are the conditional expectations knowing the trajectory of X^{m_0} starting respectively from $X^{m_0}_{s_j}$ and from $X^{m_0}_s$.

As shown on Figure 4.5 for $t_l > s_k$, $\overline{X}^{m_0,m_1}_{s_j,s,t_l}$ is defined using $X^{m_0,m_1}_{s,s} = X^{m_0}_s$ and increments from the process $X^{m_0,m_1}_{s_j,t_l}$, in contrast to $\widetilde{X}^{m_0,m_1}_{s_j,s,t_l}$ defined using $X^{m_0,m_1}_{s_j,s}$ and increments from the process X^{m_0,m_1}_{s,t_l}. Proposition 4.4.3 provides a strong formulation of a possible compromise between two error terms on the right of each inequality (4.4.6) and (4.4.7).

Proposition 4.4.3 *For any $t \in \{0, \frac{T}{2^L}, ..., T\}$, we assume U_t is $[U_t]_{Lip}$-Lipschitz. For $s_j < s < s_k$ taking their values in the discretization set $\mathcal{S}$, we define $\mathcal{K}^{m_0}_{2,s_j,s_k} = [U_{s_k}]^2_{Lip} + \mathbb{E}^{m_0}_{s_j}(|H^{m_0,\mathcal{S}}_{s_j,s_k}|^2_{d'_1})$ and $\mathcal{K}^{m_0}_{2,s,s_k} = [U_{s_k}]^2_{Lip} + \mathbb{E}^{m_0}_s(|H^{m_0,\mathcal{S}}_{s,s_k}|^2_{d'_1})$ where $|\cdot|_{d'_1}$ is the Euclidean norm on $\mathbb{R}^{d'_1}$, then*

$$\mathbb{E}^{m_0}_s\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right) \leq \mathbb{E}^{m_0}_s\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right) + \mathcal{K}^{m_0}_{2,s,s_k}\mathbb{E}^{m_0}_s\left(\left|\widetilde{X}^{m_0,m_1}_{s_j,s,s_k} - X^{m_0,m_1}_{s,s_k}\right|^2_{d'_1}\right) \tag{4.4.6}$$

and

$$\begin{aligned}\mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right) \leq & \,\mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right) \\ & + \mathcal{K}^{m_0}_{2,s_j,s_k}\mathbb{E}^{m_0}_{s_j}\left(\left|\overline{X}^{m_0,m_1}_{s_j,s,s_k} - X^{m_0,m_1}_{s_j,s_k}\right|^2_{d'_1}\right).\end{aligned} \tag{4.4.7}$$

Proof *As we simulate several independent copies of X^{m_0,m_1} (cf. the paragraph under Remark 4.2.1), we make sure that the approximations $\overline{h}$ are independent from X^{m_0,m_1}, from $\overline{X}^{m_0,m_1}$ and from $\widetilde{X}^{m_0,m_1}$ conditionally on X^{m_0}. Moreover, from definition* (4.4.5)*, $(\overline{X}^{m_0,m_1}_{s_j,s,t_l})_{t_l\geq s}$ has the same law as $(X^{m_0,m_1}_{s,t_l})_{t_l\geq s}$ and $(\widetilde{X}^{m_0,m_1}_{s_j,s,t_l})_{t_l\geq s}$ has the same law as $(X^{m_0,m_1}_{s_j,t_l})_{t_l\geq s}$. Then one can write the following*

$$\begin{aligned}\mathbb{E}^{m_0}_{s}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right) = & \quad \mathbb{E}^{m_0}_{s}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k}) - U_{s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k})\right]^2\right) \\ & \leq \mathbb{E}^{m_0}_{s}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k}) - \overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right) \\ & + \mathbb{E}^{m_0}_{s}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right) \\ & + \mathbb{E}^{m_0}_{s}\left(\left[U_{s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k})\right]^2\right)\end{aligned} \tag{4.4.8}$$

as well as

$$\begin{aligned}\mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right) = & \quad \mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k}) - U_{s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k})\right]^2\right) \\ & \leq \mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k}) - \overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right) \\ & + \mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right) \\ & + \mathbb{E}^{m_0}_{s_j}\left(\left[U_{s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k})\right]^2\right)\end{aligned} \tag{4.4.9}$$

which yield (4.4.6) *and* (4.4.7)*.*

Proposition 4.4.3 requires Lipschitz property of U which is fulfilled if f is Lipschitz. Using similar steps to the one presented in [99], we show in Lemma 4.4.1 this Lipschitz property for (ODP). Using similar arguments, one can also show this property for (Snl) if g is also Lipschitz. We point out that another option is the one based on differentiability assumptions as in [92].

Consider the following extension of (4.4.3) with a driver f that depends also on X

$$\begin{cases}\widehat{Y}_{s_k} & = & \mathbb{E}_{s_k}\left(\widehat{Y}_{\delta(s_k)} + \Delta_{s_k} f_{s_k}(X_{s_k}, \widehat{Y}_{\delta(s_k)}, \widehat{Z}_{s_k})\right) \\ \widehat{Z}_{s_k} & = & \frac{1}{\Delta_{s_k}}\mathbb{E}_{s_k}\left(\widehat{Y}_{\delta(s_k)}(W_{\delta(s_k)} - W_{s_k})\right) = \frac{1}{\sqrt{\Delta_{s_k}}}\mathbb{E}_{s_k}\left(\widehat{Y}_{\delta(s_k)}\theta_{\delta(s_k)}\right)\end{cases}$$

where $\theta_{\delta(s_k)} \sim \mathcal{N}(0, I_{d_1})$. Replacing s_k by k and using Markov property, $\widehat{Y}_{s_k} = y_k(X_{s_k})$ and $\widehat{Z}_{s_k} = z_k(X_{s_k})$ (cf [67]) with

$$\begin{cases}y_k(x) & = & \mathbb{E}\left(y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) + \Delta_k f_k(x, y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})), z_k(x))\right) \\ z_k(x) & = & \frac{1}{\sqrt{\Delta_k}}\mathbb{E}\left(y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}))\theta_{k+1}\right).\end{cases}$$

Lemma 4.4.1 *Assume that $f(t,x,y,z)$ is $[f]_{Lip}$-Lipschitz continuous with respect to x, y and z uniformly in $t\in[0,T]$, for the particular case $f(T,x)$ we denote by $[f_T]_{Lip}$ the Lipschitz coefficient. The coefficients $b(t,x)$ and $\sigma(t,x)$ of the Markov process 4.4.1 are also assumed Lipschitz continuous in x uniformly with respect to $t\in[0,T]$ with Lipschitz coefficients denoted $[b]_{Lip}$ and $[\sigma]_{Lip}$. Assume that $n\geqslant n_0$ (in order to provide sharper constants depending on $n_0\geqslant 1$).*

Then for every $k\in\{0,....,n-1\}$, y_k is $[y_k]_{Lip}$-Lipschitz continuous with

$$[y_k]_{Lip}\leqslant\left([y_{k+1}]_{Lip}\,e^{\Delta_k C'}+\Delta_k\,[f]_{Lip}\right)$$

where $C'=[b]_{Lip}+\frac{1}{2}\left([\sigma]_{Lip}^2+\frac{T}{n_0}[b]_{Lip}^2\right)+[f]_{Lip}\left(1+\sqrt{d_1}\sqrt{\frac{n_0}{T}}\right)$.

Moreover the functions z_k are $[z_k]_{Lip}$-Lipschitz continuous with

$$[z_k]_{Lip}\leqslant\frac{1}{\sqrt{\Delta_k}}\,[y_{k+1}]_{Lip}\,e^{\Delta_k C_{b,\sigma,T}}\sqrt{d_1}.$$

If $\Delta_k=h$ is homogeneous with respect to k, we have

$$[y_k]_{Lip}\leqslant\left([f_T]_{Lip}+[f]_{Lip}\,\frac{T}{n_0}C'\right)e^{C'(T-t_k)}$$

and

$$[z_k]_{Lip}\leqslant\sqrt{d_1}\sqrt{\frac{n_0}{T}}\left([f_T]_{Lip}-C'\,[f]_{Lip}\right)e^{C'(T-t_k)}e^{\frac{T}{n_0}C_{b,\sigma,T}}.$$

Proof *Assume by backward induction that y_{k+1} is $[y_{k+1}]_{Lip}$-Lipschitz continuous. For every $x,x'\in\mathbb{R}^{d_1}$, we have*

$$\begin{aligned}
y_k(x)-y_k(x')&=\mathbb{E}\left[y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}))-y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))\right]\\
&+\Delta_k\mathbb{E}\left[f_k(x,y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})),z_k(x))-f_k(x',y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})),z_k(x))\right]\\
&+\Delta_k\mathbb{E}\left[f_k(x',y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})),z_k(x))-f_k(x',y_{k+1}(\mathcal{E}_k(x',\theta_{k+1})),z_k(x))\right]\\
&+\Delta_k\mathbb{E}\left[f_k(x',y_{k+1}(\mathcal{E}_k(x',\theta_{k+1})),z_k(x))-f_k(x',y_{k+1}(\mathcal{E}_k(x',\theta_{k+1})),z_k(x'))\right].
\end{aligned}$$

and

$$z_k(x)-z_k(x')=\frac{1}{\sqrt{\Delta_k}}\mathbb{E}\left((y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}))-y_{k+1}(\mathcal{E}_k(x',\theta_{k+1})))\theta_{k+1}\right).$$

We denote

$$\begin{aligned}
A_{x,x'}&=\frac{[f_k(x,y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})),z_k(x))-f_k(x',y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}),z_k(x))]\,1_{\{\mathcal{A}_{x,x'}\neq 0\}}}{\mid x-x'\mid_{d_1}}\\
B_{x,x'}&=\frac{[f_k(x',y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}),z_k(x))-f_k(x',y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}),z_k(x))]\,1_{\{\mathcal{B}_{x,x'}\neq 0\}}}{y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}))-y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))}\\
C_{x,x'}&=\frac{[f_k(x,y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}),z_k(x))-f_k(x',y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}),z_k(x))]\,1_{\{\mathcal{C}_{x,x'}\neq 0\}}}{\mid z_k(x)-z_k(x')\mid_{d_1}}
\end{aligned}$$

where $| \ x \ |_{d_1}= \sqrt{x_1^2+....+x_{d_1}^2}$, $\mathcal{A}_{x,x} =| \ x-x' \ |_{d_1}$, $\mathcal{B}_{x,x} = y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) - y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))$ *and* $\mathcal{C}_{x,x} =| \ z_k(x)-z_k(x') \ |_{d_1}$.

We have

$$\begin{aligned} y_k(x)-y_k(x') &= \mathbb{E}\left[\left(y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}))-y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))\right)(1+B_{x,x'}\Delta_k)\right] \\ &+\Delta_k\mathbb{E}\left[\frac{1}{\sqrt{\Delta_k}}C_{x,x'}\mathbb{E}\left(| \ (y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}))-y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))).\theta_{k+1} \ |_{d_1}\right)\right] \\ &+\Delta_k\mathbb{E}\left[A_{x,x'} \ | \ x-x' \ |_{d_1}\right]. \end{aligned}$$

Using the Lipschitz property of f_k *and* y_{k+1} *we have*

$$\begin{aligned} | \ y_k(x)-y_k(x') \ | &\leqslant [y_{k+1}]_{Lip}(1+[f]_{Lip}\Delta_k)\mathbb{E}\left[| \ \mathcal{E}_k(x,\theta_{k+1})-\mathcal{E}_k(x',\theta_{k+1}) \ |\right] \\ &+\Delta_k[f]_{Lip}\frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip}\sqrt{\sum_{i=1}^{d_1}\mathbb{E}\left[(\mathcal{E}_k(x,\theta_{k+1})-\mathcal{E}_k(x,\theta_{k+1}))\theta^i_{k+1}\right]^2} \\ &+\Delta_k\left[f\right]_{Lip} | \ x-x' \ |_{d_1} \ . \end{aligned}$$

By applying Cauchy-Schwarz's inequality and knowing that $\mathbb{E}((\theta^i_{k+1})^2)=1$, *for* $i\in\{1,...,d_1\}$, *we have*

$$\begin{aligned} | \ y_k(x)-y_k(x') \ | &\leqslant [y_{k+1}]_{Lip}\left(1+\Delta_k[f]_{Lip}+\Delta_k[f]_{Lip}\tfrac{1}{\sqrt{\Delta_k}}\sqrt{d_1}\right) \\ &\times \sqrt{\underbrace{\mathbb{E}\left[\left(\mathcal{E}_k(x,\theta_{k+1})-\mathcal{E}_k(x',\theta_{k+1})\right)^2\right]}_{D_{x,x'}}} \\ &+ \Delta_k\left[f\right]_{Lip} | \ x-x' \ |_{d_1} \ . \end{aligned}$$

As $b_k(.)$ *and* $\sigma_k(.)$ *are Lipschitz, by elementary computations already carried out in [27, 99], we have*

$$\begin{aligned} D_{x,x'} &= \mathbb{E}\left(\left[\mathcal{E}_k(x,\theta_{k+1})-\mathcal{E}_k(x',\theta_{k+1})\right]^2\right) \\ &= \mathbb{E}\left(\left[x-x'+\Delta_k\left[b_k(x)-b_k(x')\right]+\sqrt{\Delta_k}\theta_{k+1}\left[\sigma_k(x)-\sigma_k(x')\right]\right]^2\right) \\ &\leqslant \left(1+\Delta_k(2\left[b\right]_{Lip}+\left[\sigma\right]^2_{Lip}+\Delta_k\left[b\right]^2_{Lip})\right) | \ x-x' \ |^2_{d_1} \\ &\leqslant (1+\Delta_kC_{b,\sigma,T})^2 \ | \ x-x' \ |^2_{d_1} \\ &\leqslant e^{2\Delta_kC_{b,\sigma,T}} \ | \ x-x' \ |^2_{d_1} \end{aligned}$$

where $C_{b,\sigma,T}$ *can be taken equal to* $[b]_{Lip}+\frac{1}{2}\left(\left[\sigma\right]^2_{Lip}+\frac{T}{n_0}\left[b\right]^2_{Lip}\right)$.
This brings us to,

$$\begin{aligned} | \ y_k(x)-y_k(x') \ | &\leqslant \left(\Delta_k\left[f\right]_{Lip}+[y_{k+1}]_{Lip}\left(1+\Delta_k[f]_{Lip}\left(1+\sqrt{\frac{d_1}{\Delta_k}}\right)\right)e^{\Delta_kC_{b,\sigma,T}}\right) | \ x-x' \ |_{d_1} \\ &\leqslant \left(\Delta_k\left[f\right]_{Lip}+[y_{k+1}]_{Lip}e^{\Delta_k(C_{b,\sigma,T}+C_{f,d_1,T})}\right) | \ x-x' \ |_{d_1} \end{aligned}$$

where $C_{f,d_1,T}$ is taken equal to $[f]_{Lip}\left(1+\sqrt{d_1}\sqrt{\frac{n_0}{T}}\right)$.

We conclude that y_k is Lipschitz continuous with coefficient $[y_k]_{Lip}$ satisfying

$$[y_k]_{Lip} \leqslant \left(\Delta_k [f]_{Lip} + [y_{k+1}]_{Lip} e^{\Delta_k C'}\right)$$

where $C' = C_{b,\sigma,T} + C_{f,d_1,T}$.
Moreover, using that $\theta_{k+1} \sim \mathcal{N}(0, I_{d_1})$, combined with Cauchy-Schwarz's inequality and Lipschitz property we get

$$\begin{aligned}
\mid z_k(x) - z_k(x') \mid_{d_1} &\leqslant \frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip}\sqrt{\sum_{i=1}^{d_1}\mathbb{E}\left[(\mathcal{E}_k(x,\theta_{k+1}) - \mathcal{E}_k(x,\theta_{k+1}))\theta^i_{k+1}\right]^2} \\
&\leqslant \frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip}\sqrt{\sum_{i=1}^{d_1}\mathbb{E}\left[(\mathcal{E}_k(x,\theta_{k+1}) - \mathcal{E}_k(x,\theta_{k+1}))\right]^2\mathbb{E}\left[\theta^i_{k+1}\right]^2} \\
&\leqslant \frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip}\sqrt{d_1}\sqrt{D_{x,x'}} \\
&\leqslant \frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip}\sqrt{d_1}e^{\Delta_k C_{b,\sigma,T}} \mid x - x' \mid_{d_1}.
\end{aligned}$$

Thus, z_k is Lipschitz continuous with coefficient $[z_k]_{Lip}$ satisfying

$$[z_k]_{Lip} \leqslant \frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip} e^{\Delta_k C_{b,\sigma,T}}\sqrt{d_1}.$$

Assuming homogeneous time increment $\Delta_k = h$, we have

$$e^{C'kh}[y_k]_{Lip} \leqslant [y_{k+1}]_{Lip} e^{C'(k+1)h} + e^{C'kh}[f]_{Lip} h.$$

which yields

$$\begin{aligned}
e^{C'kh}[y_k]_{Lip} &\leqslant [f_T]_{Lip} e^{C'nh} + [f]_{Lip} h \sum_{l=k}^{n-1} e^{C'lh} \\
&\leqslant [f_T]_{Lip} e^{C'T} + [f]_{Lip} h \frac{e^{C'T} - e^{C'kh}}{e^{C'T} - 1} \\
&\leqslant [f_T]_{Lip} e^{C'T} + [f]_{Lip} h \frac{e^{C'T}}{e^{C'T} - 1} \\
&\leqslant [f_T]_{Lip} e^{C'T} + [f]_{Lip} hC'e^{C'T}.
\end{aligned}$$

Finally we have

$$[y_k]_{Lip} \leqslant [f_T]_{Lip} e^{C'(T-s_k)} + [f]_{Lip} hC'e^{C'(T-s_k)}. \tag{4.4.10}$$

and

$$[z_k]_{Lip} \leqslant \frac{1}{\sqrt{h}}([f_T]_{Lip} - C'[f]_{Lip})e^{C'(T-t_k)}e^{hC_{b,\sigma,T}}\sqrt{d_1}.$$

4.5 Some numerical results

In this section we test the above conditional MC learning procedure on various examples including BSDE, American option and risk measure. The fact that the driver f depends also on X is not a burden to the use of our method. All simulations are run on a laptop that has an Intel i7-7700HQ CPU and a single GeForce GTX 1060 GPU programmed with the CUDA/C application programming interface. We refer the reader to [94] for an introduction to CUDA programming.

4.5.1 Allen-Cahn equation

We consider (ODP) simulation as presented in Section 4.3.2, we use the following functions

$$f(t,x,y,z) = y - y^3,$$
$$f(T,x) = \left[2 + \frac{2}{5}|x|^2_{d_1}\right]^{-1}$$

and

$$\mathcal{E}_{t_k}(x,w) = x + \sqrt{2}w, \quad X_{t_0} = 0.$$

We would like to approximate the solution $u(t,x)$ of the Allen-Cahn PDE defined as follows, $u(T,x) = f(T,x)$,

$$\frac{\partial u}{\partial t}(t,x) + u(t,x) - [u(t,x)]^3 + (\Delta_x u)(t,x) = 0. \tag{4.5.1}$$

A benchmark approximation $u_b(0,x)$ for the solution $u(0,x)$ of the PDE (4.5.1) is given in [Section 4.2 ; [25]].

Table 4.1 shows the solution $u(0,0)$ of equation (4.5.1), calculated by learned and simulated expression, with respect to the number of inner trajectories M_1. The benchmark solution $u_b(0,0)$ is equal to 0.0528 for $T = 0.3$ and $d_1 = 100$. The standard deviation of each expression and the runtime in seconds are also given. We reduce the bias by increasing the number of inner trajectories. Table 4.1 shows that a relative small number of outer and inner trajectories is sufficient to observe a small variance and bias for both options. In fact, we show that the standard deviation is already acceptable for $M_0 = 2^4$ outer trajectories and the bias is acceptable for $M_1 = 2^6$ inner trajectories with an execution time of 56 millisecond.

Table 4.1 – Numerical simulations for PDE (4.5.1) : $T = 0.3$, $M_0 = 2^4$, $d_1 = 100$, $L = 4$; [Benchmarck solution] $u_b(0,0) = 0.0528$.

M_1	Learned		Simulated		Runtime in sec. (10^{-3})
	Y_0^{learn}	std	Y_0^{sim}	std	
2^4	0.0454	(± 0.0093)	0.0455	(± 0.0073)	13
2^5	0.0513	(± 0.0011)	0.0517	(± 0.0008)	23
2^6	0.0523	(± 0.0004)	0.0518	(± 0.0006)	56
2^7	0.0526	(± 0.0003)	0.0515	(± 0.0001)	119
2^8	0.0525	(± 0.0002)	0.0517	(± 0.0002)	227
2^9	0.0527	(± 0.0002)	0.0515	(± 0.0002)	414

Table 4.2 shows the solution $u(0,0)$ of equation (4.5.1), calculated by learned

and simulated expression, with respect to the number of inner trajectories M_1, for a long time horizon ($T = 1$). The benchmark solution is equal to 0.0338 for $T = 1$, $d_1 = 100$. To achieve a similar level of variance and bias we need more outer and inner trajectories than in Table 4.1. In fact for $M_0 = 2^5$ of outer trajectories and $M_1 = 2^6$ of inner trajectories we obtained an acceptable bias and standard deviation in 4 seconds.

Table 4.2 – Numerical simulations for PDE (4.5.1) : $T = 1$, $d_1 = 100$, $M_0 = 2^5$, $L = 6$; [Benchmarck solution] $u_b(0, 0) = 0.0338$.

M_1	Learned		Simulated		Runtime in sec.
	Y_0^{learn}	std	Y_0^{sim}	std	
2^5	0.0345	(± 0.0008)	0.0350	(± , 0.0021)	2
2^6	0.0333	(± 0.0003)	0.0326	(± 0.0004)	4
2^7	0.0334	(± 0.0002)	0.0330	(± 0.0003)	7
2^8	0.0336	(± 0.0002)	0.0332	(± 0.0002)	12
2^9	0.0336	(± 0.0001)	0.0331	(± 0.0001)	27

4.5.2 Multidimensional Burgers-type PDEs with explicit solution

We assume the (ODP) setting presented in Section 4.3.2, we use the following functions

$$f(t, x, y, z) = \left(y - \frac{2 + d_1}{2d_1}\right)\left(\sum_{i=1}^{d_1} z_i\right),$$

$$f(T, x) = \frac{\exp\left(T + \frac{1}{d_1}\sum_{i=1}^{d_1} x_i\right)}{1 + \exp\left(T + \frac{1}{d_1}\sum_{i=1}^{d_1} x_i\right)}$$

and

$$\mathcal{E}_{t_k}(x, w) = x + \frac{d_1}{\sqrt{2}}w, \quad X_{t_0} = 0.$$

We simulate the solution $u(t, x)$ of the multidimensional Burgers-type PDE (cf [40], Example 4.6) defined as follows, $u(T, x) = f(T, x)$,

$$\frac{\partial u}{\partial t}(t, x) + \frac{{d_1}^2}{2}(\Delta_x u)(t, x) + \left(u(t, x) - \frac{2 + d_1}{2d_1}\right)\left(d_1\sum_{i=1}^{d_1}\frac{\partial u}{\partial x_i}(t, x)\right) = 0. \quad (4.5.2)$$

PDE (4.5.2) admits an explicit solution, we refer the reader to [Lemma 4.3, [25]] for more details. The value of the solution $u(0, 0)$ is 0.5000 for $T = 0.2$ and $d_1 = 100$.

Table 4.3 shows the solution $u(0, 0)$ of the equation (4.5.2), calculated by learned and simulated expression, with respect to the number of inner trajectories M_1. The approximation of the standard deviation of each expression and the runtime in seconds are also given. We show that the standard deviation of both results should be reduced by increasing the number of outer trajectories.

TABLE 4.3 – Numerical simulations for PDE (4.5.2) : $T = 0.2$, $d_1 = 100$, $M_0 = 2^6$, $L = 5$; [Explicit solution] $u(0, 0) = 0.5000$.

M_1	Learned		Simulated		Runtime in sec.
	Y_0^{learn}	std	Y_0^{sim}	std	
2^8	0.4785	(± 0.0428)	0.0517	(± 0.0431)	7
2^9	0.5113	(± 0.0450)	0.5108	(± 0.0450)	16
2^{10}	0.4966	(± 0.0448)	0.4912	(± 0.0447)	27
2^{11}	0.5022	(± 0.0421)	0.5012	(± 0.0435)	49

In Table 4.4 we show the computed solution of the equation (4.5.2), calculated by learned and simulated expression with respect to the number of outer trajectories M_0. The standard deviation of each expression and the runtime in seconds are also given. We reduce the standard deviation by increasing the number of outer trajectories.

TABLE 4.4 – Numerical simulations for PDE (4.5.2) : $T = 0.2$, $d_1 = 100$, $M_1 = 2^{11}$, $L = 5$; [Explicit solution] $u(0, 0) = 0.5000$.

M_0	Learned		Simulated		Runtime in sec.
	Y_0^{learn}	std	Y_0^{sim}	std	
2^5	0.4953	(± 0.0618)	0.4941	(± 0.0615)	24
2^6	0.5022	(± 0.0424)	0.501284	(± 0.0435)	49
2^7	0.5079	(± 0.0346)	0.5066	(± 0.0342)	103
2^8	0.5158	(± 0.0221)	0.5151	(± 0.0221)	194
2^9	0.5023	(± 0.0164)	0.5029	(± 0.0164)	408

4.5.3 Time-dependent reaction-diffusion-type example PDEs with oscillating explicit solutions

Let $\kappa = 0.6$, $\lambda = \frac{1}{\sqrt{d_1}}$, we use the following functions

$$f(t, x, y, z) = \min\left\{1, \left[y - \kappa - 1 - \sin\left(\lambda \sum_{i=1}^{d_1} x_i\right) \exp\left(\frac{\lambda^2 d_1 (t - T)}{2}\right)\right]^2\right\},$$

$$f(T, x) = 1 + \kappa + \sin\left(\lambda \sum_{i=1}^{d_1} x_i\right)$$

and

$$\mathcal{E}_{t_k}(x, w) = x + w, \quad X_{t_0} = 0.$$

We simulate the solution $u(t, x)$ of the time dependent reaction-diffusion-type PDE (cf [67], Section 6.1) defined as follows, $u(T, x) = f(T, x)$,

$$\frac{\partial u}{\partial t}(t, x) + \min\left\{1, \left[y - \kappa - 1 - \sin\left(\lambda \sum_{i=1}^{d_1} x_i\right) \exp\left(\frac{\lambda^2 d_1 (t - T)}{2}\right)\right]^2\right\} + \frac{1}{2}(\Delta_x u)(t, x) = 0. \tag{4.5.3}$$

The explicit solution of the PDE (4.5.3) is given in [Lemma 4.4 ; [25]].

Table 4.5 shows the approximated solution of the equation (4.5.3), calculated by learned and simulated expression, with respect to the number of inner trajectories

M_1. The standard deviation of each expression, and the runtime in seconds are also given. The benchmark solution is equal to 1.6000 for $T = 1$, $d_1 = 100$.

TABLE 4.5 – Numerical simulations for PDE (4.5.3) : $T = 0.5$, $d_1 = 100$, $M_0 = 2^{10}$, $L = 3$; [Benchmark solution] $u_b(0,0) = 1.6000$.

M_1	Learned		Simulated		Runtime in sec. (10^{-3})
	Y_0^{learn}	std	Y_0^{sim}	std	
2^5	1.8197	(± 0.0386)	1.7587	(± 0.0287)	244
2^6	1.7125	(± 0.0104)	1.6799	(± 0.0116)	311
2^7	1.6605	(± 0.0037)	1.6376	(± 0.0091)	466
2^8	1.6458	(± 0.0023)	1.6290	(± 0.0089)	817
2^9	1.6439	(± 0.0019)	1.6283	(± 0.0061)	1526

4.5.4 A Hamilton-Jacobi-Bellman (HJB) equation

We assume here the driver to be equal to

$$f(t,x,y,z) = -|z|^2_{d_1},$$

$$f(T,x) = ln\left(\frac{1}{2}\left[1 + |x|^2_{d_1}\right]\right)$$

and

$$\mathcal{E}_{t_k}(x,w) = x + \sqrt{2}w, \quad X_{t_0} = 0.$$

We calculate the solution $u(t,x)$ of the HJB equation (cf [41] Section 4.2) defined by $u(T,x) = f(T,x)$,

$$\frac{\partial u}{\partial t}(t,x) + (\Delta_x u)(t,x) - |(\nabla_x u)(t,x)|^2_{d_1} = 0. \tag{4.5.4}$$

PDE (4.5.4) admits a benchmark solution. We refer the reader to [Lemma 4.2; [25]] for more details.

In Figure 4.6 we show the difference between $Y_{s_k} = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}}$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}} + (\delta(s_k) - s_k)f(s_k, \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}}, \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})\right)$ with respect to the discretization time steps. On the left, we perform the conditional MC procedure taking $\overline{s_k} = T$. On the right, we perform the procedure with the bias control presented in Section 4.2.3 , taking $\overline{s_k} = \left(s_k + \frac{3}{8}\right) \wedge T$ with $s_k \in \mathcal{S}^{i^*} = \{0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$. We show that the control allows to reduce the bias propagation.

Figure 4.7 shows the convergence of the learned and simulated expression to the benchmark value with respect to the number of inner trajectories. In particular, we observe that the both expressions converge to the benchmark solution with a small variance when $M_1 = 2^{17}$.

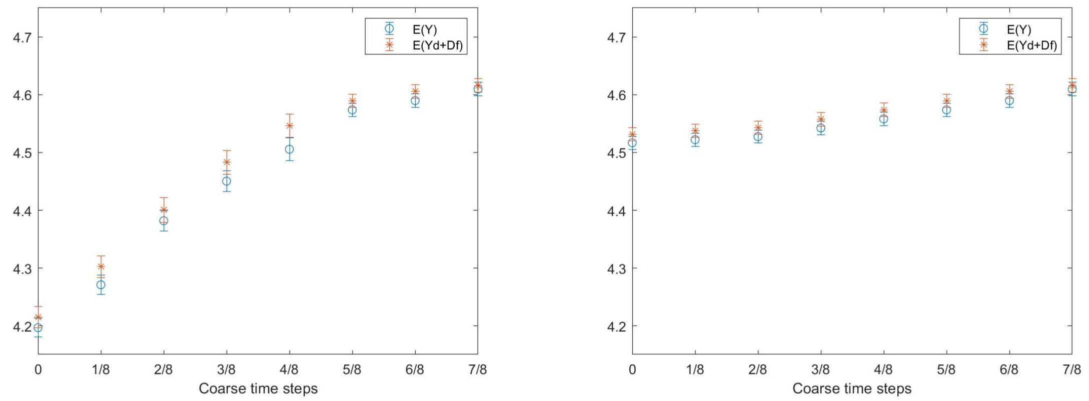


FIGURE 4.6 – Y_{s_k} vs. $\displaystyle\frac{1}{M_0}\sum_{m_0=1}^{M_0}\Big(\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}} + (\delta(s_k)-s_k) f(s_k, \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}}, \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})\Big)$ [Left] $\overline{s_k} = T$ without bias control, [Right] $\overline{s_k} = \big(s_k + \frac{3}{8}\big) \wedge T$ with bias control : $T = 1$, $d_1 = 100$, $M_0 = 2^7$, $M_1 = 2^{15}$, $L = 3$.

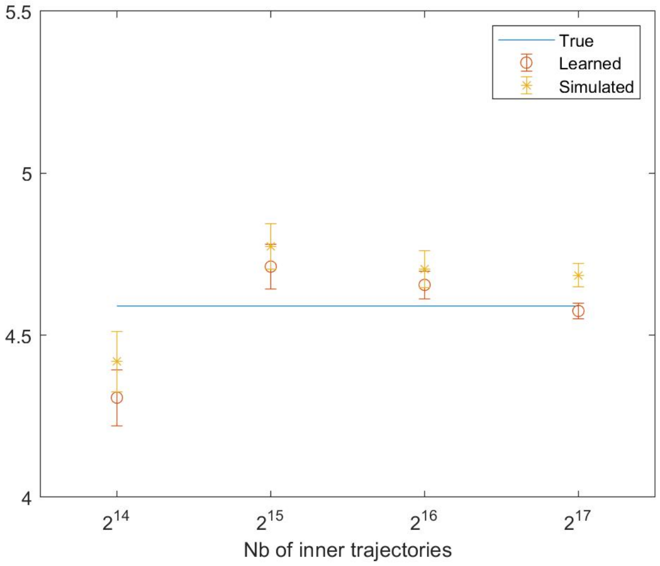


FIGURE 4.7 – Numerical solution of PDE (4.5.4) calculated by learned and simulated expression : $T = 1$, $d_1 = 100$, $M_0 = 2^7$, $L = 3$.

4.5.5 Pricing of European financial derivatives with different interest rates for borrowing and lending

Assuming $\mu = 0.06$, $\sigma = 0.2$, $R^l = 0.04$ and $R^b = 0.06$, we introduce the following functions

$$f(t,x,y,z) = -R^l y - \frac{(\mu - R^l)}{\sigma}\sum_{i=1}^{d_1} z_i + (R^b - R^l)\max\{0, \frac{1}{\sigma}\sum_{i=1}^{d_1} z_i - y\},$$

$$f(T,x) = \max\{\max_{1\le i\le d_1} x_i - 120, 0\} - 2\max\{\max_{1\le i\le d_1} x_i - 150, 0\}$$

and

$$\mathcal{E}_{t_k}(x, w) = x \exp\left((\mu - \frac{\sigma^2}{2})\Delta_t + \sigma w\right), \quad X_{t_0} = 100.$$

Let u defined as the solution of the following PDE, $u(T, x) = f(T, x)$,

$$\frac{\partial u}{\partial t}(t,x) + \frac{\sigma}{2}\sum_{i=1}^{d_1}|x_i|^2\frac{\partial^2 u}{\partial x_i^2}(t,x) \tag{4.5.5}$$
$$+\max\{R^b\left(\sum_{i=1}^{d_1}x_i\left(\frac{\partial u}{\partial x_i}(t,x)\right) - u(t,x)\right), R^l\left(\sum_{i=1}^{d_1}x_i\left(\frac{\partial u}{\partial x_i}(t,x)\right) - u(t,x)\right)\} = 0.$$

PDE (4.5.5) has a benchmark solution given in [Section 4.4 ; [25]]. This benchmark solution is equal to 21.299 for $T = 0.5$ and $d_1 = 100$.

Figure 4.8 shows the approximation of the solution of PDE (4.5.5), calculated by learned and simulated expression, with respect to the number of inner trajectories. We show that 2^7 outer trajectories and 2^{11} inner trajectories are sufficient to get an accurate approximation of the solution as the obtained values are in the corridor of the standard deviation of the benchmark solution. No bias cut is needed here. The runtime with 2^7 outer trajectories and 2^{11} inner trajectories is 53 seconds.

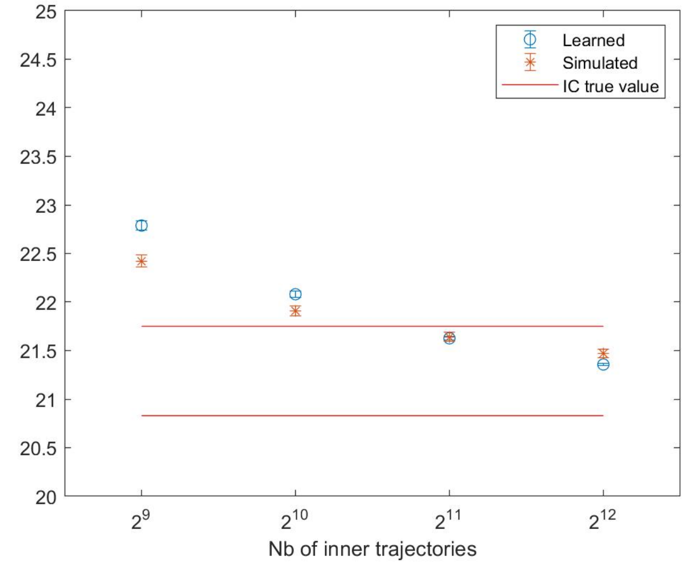


FIGURE 4.8 – Numerical solution of PDE (4.5.5) calculated by learned and simulated expression : $T = 0.5$, $d_1 = 100$, $M_0 = 2^7$, $L = 2$.

4.5.6 A PDE example with quadratically growing derivatives and an explicit solution

Assuming the (ODP) setting presented in Section 4.3.2, let $\alpha = 0.4$ and $\psi(t, x) = \sin\left(\left[T - t + |x|^2_{d_1}\right]^\alpha\right)$, we introduce the following functions,

$$f(t, x, y, z) = |z|^2_{d_1} - |\nabla_x\psi(t, x)|^2_{d_1} - \frac{\partial\psi}{\partial t}(t, x) - \frac{1}{2}(\Delta_x\psi)(t, x),$$

$$f(T,x) = \sin\left(|x|^{2\alpha}_{d_1}\right)$$

and

$$\mathcal{E}_{t_k}(x,w) = x + w, \quad X_{t_0} = 0.$$

Let u defined as the solution of the following PDE, $u(T,x) = f(T,x)$,

$$\begin{aligned} \tfrac{\partial u}{\partial t}(t,x) + |\nabla_x u(t,x)|^2_{d_1} + \tfrac{1}{2}(\Delta_x u)(t,x) &= \qquad \tfrac{\partial \psi}{\partial t}(t,x) \\ &+ \quad |\nabla_x \psi(t,x)|^2_{d_1} + \tfrac{1}{2}(\Delta_x \psi)(t,x). \end{aligned} \tag{4.5.6}$$

Straight use of Itô's Lemma shows that PDE (4.5.6) has an explicit solution $u(t,x) = \psi(t,x)$, we refer the reader to [Section 6.1 ; [67]] for more details.

Figure 4.9 is related to Propostion 4.4.3 that controls the error of regressions with different starting points. Here we prefered to show the distributions rather than the quadratic error which is small. On the left of Figure 4.9 we have the "Trained" value $Y^{m_0}_{\frac{249}{256}} = \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{249}{256},T}$ and the "Tested" $\dfrac{1}{M_1}\displaystyle\sum_{m_1=1}^{M_1}\left(\overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{248}{256},\frac{250}{256}}\left(X^{m_0,m_1}_{\frac{249}{256},\frac{250}{256}}\right) + \Delta_s f\left(\dfrac{249}{256}, \overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{248}{256},\frac{250}{256}}\left(X^{m_0,m_1}_{\frac{249}{256},\frac{250}{256}}\right), \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{\frac{249}{256},\frac{250}{256}}\right)\right)$ with $\mathcal{S}^{i^*} \in \{0, \frac{1}{256}, \frac{2}{256}, ..., 1\}$. On the right we show the "Trained" $Y^{m_0}_{\frac{253}{256}} = \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{253}{256},T}$ and the "Tested" $\dfrac{1}{M_1}\displaystyle\sum_{m_1=1}^{M_1}\left(\overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{252}{256},\frac{254}{256}}\left(X^{m_0,m_1}_{\frac{253}{256},\frac{254}{256}}\right) + \Delta_s f\left(\dfrac{253}{256}, \overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{252}{256},\frac{254}{256}}\left(X^{m_0,m_1}_{\frac{253}{256},\frac{254}{256}}\right), \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{\frac{253}{256},\frac{254}{256}}\right)\right)$ at a different time step 254/256. Figure 4.9 shows very similar distributions which strengthen the benefit of our trick.

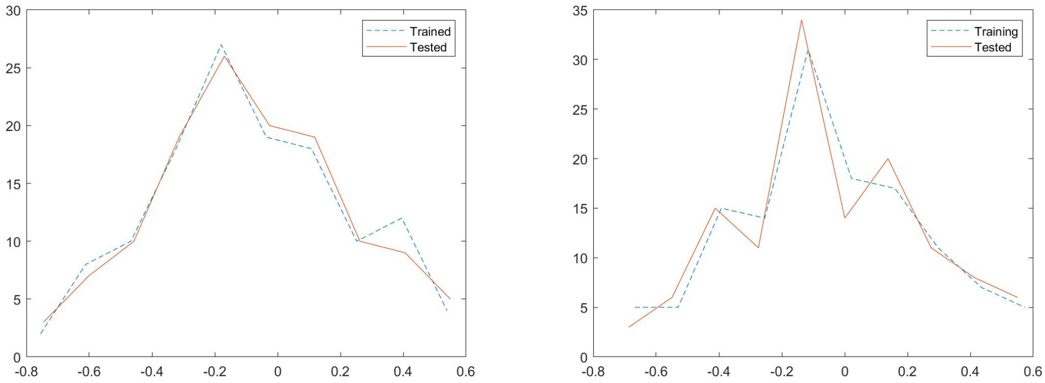


FIGURE 4.9 – [Left] Distribution of $Y^{m_0}_{\frac{249}{256}}$ called "Trained" vs. its different starting point approximation called "Tested" [Right] $Y^{m_0}_{\frac{253}{256}}$ called "Trained" vs. its different starting point approximation called "Tested" : $T = 1$, $d = 100$, $M_0 = 2^7$, $M_1 = 2^{12}$, $L = 8$.

Figure 4.10 shows the numerical solution of PDE (4.5.6), calculated by learned and simulated expression, with respect to different number of coarse time step. We show that $L = 8$ is sufficient to discretize the problem when the time horizon T is equal to 1. This convergence is achieved in 620 seconds of runtime.

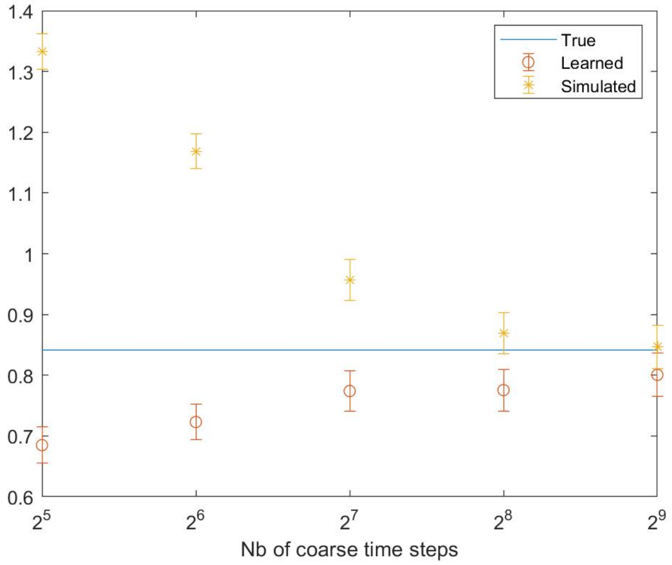


FIGURE 4.10 – Numerical solution of PDE (4.5.6) calculated by learned and simulated expression with a bias control : $T = 1$, $d_1 = 100$, $M_0 = 2^7$, $M_1 = 2^7$.

4.5.7 American geometric put option

Given the (Snl) setting of Section 4.3.3 with a driver $f = 0$, we consider an American geometric put option with constant interest rate r and a payoff

$$g(x) = \left[K - \prod_{i=1}^{d_1} (x_i)^{1/d_1} \right]^+ \qquad (4.5.7)$$

with an asset X given by $X_t^i = X_s^i \exp\Big((r - \frac{\sigma^2}{2})(t-s) + \sigma(W_t^i - W_s^i) \Big)$, $t > s$, $1 \le i \le d_1$, $r = \log(1.1)$, $\sigma = 0.4$, $K = X_0^i = 100$ and $d_1 = 20$.

We approximate the price V_0 associated to payoff (4.5.7). We choose the dimension $d_1 = 20$ to make sure that the variance of the problem is sufficiently large. We point out however that it works well for $d_1 = 100$.

In Table 4.6 we show the price of an American geometric put option, calculated by simulated expression V_0^{sim}, for different maturities. Indeed, V_0^{learn} provides almost the same values. From top to bottom we have : a variance adjustment [VA], a bias control [BC] and a combination of [BC] and [VA]. We show that the simulated expression with a combination of [BC] and [VA] gives a good approximation of the price even for long maturity $T = 2$. However, one needs to use variance adjustment that is important for events on the exercise frontier as well as bias control to cut the propagation of bias.

TABLE 4.6 – Numerical simulations for American option (4.5.7) simulated formula, [BC] bias control [VA] variance adjustment : $d_1 = 20$, $M_0 = 2^{11}$, $M_1 = 2^{12}$.

	L = 2 (T = 0.5)	L = 3 (T = 1)	L = 4 (T = 2)
[VA]	2.561 (± 0.035)	4.236 (± 0.042)	6.363 (± 0.054)
[BC]	2.493 (± 0.041)	3.734 (± 0.061)	5.130 (± 0.089)
[VA] + [BC]	2.291 (± 0.035)	2.890 (± 0.037)	3.961 (± 0.055)
Real Price	2.153	2.871	3.754

Figure 4.11 shows the difference between $\frac{1}{M_0}\sum_{m_0=1}^{M_0} e^{-r(\delta(s_k)-s_k)}V^{m_0}_{\delta(s_k)} = \frac{1}{M_0}\sum_{m_0=1}^{M_0} e^{-r(\delta(s_k)-s_k)}(\widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}})$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}(\widetilde{w}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})$ with respect to the time discretization. On the left, we perform the conditional MC procedure by taking $\overline{s_k} = T$. On the right, we perform the procedure with the bias control presented in section 4.2.3 by taking $\overline{s_k} = \left(s_k + \frac{3}{8}\right) \wedge T$ with $s_k \in \mathcal{S}^{i^*} = \{0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$. We show that the control allows to reduce the bias propagation.

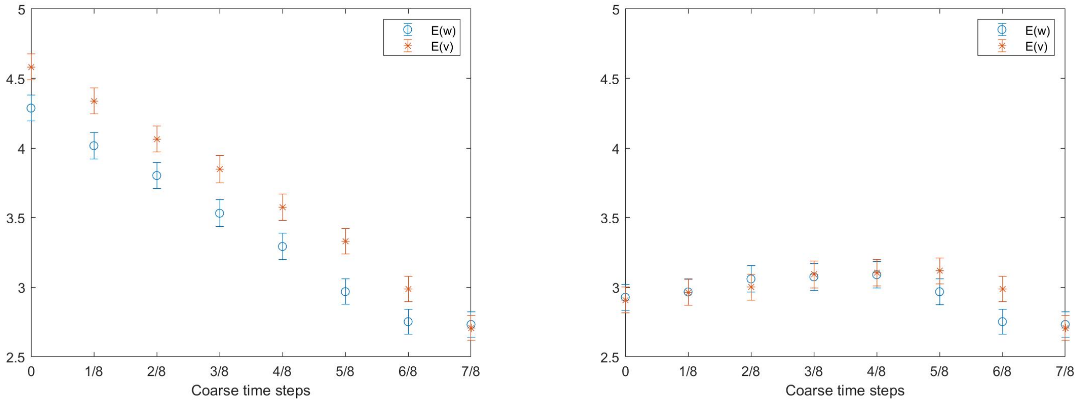


FIGURE 4.11 – $\frac{1}{M_0}\sum_{m_0=1}^{M_0} e^{-r(\delta(s_k)-s_k)}V^{m_0}_{\delta(s_k)}$ vs. $\frac{1}{M_0}\sum_{m_0=1}^{M_0}(\widetilde{w}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})$; [Left] $\overline{s_k} = T$ without bias control [Right] $\overline{s_k} = \left(s_k + \frac{3}{8}\right) \wedge T$ with bias control : $d_1 = 20$, $M_0 = 2^{11}$, $M_1 = 2^{12}$, $T = 1$ and $L = 3$.

Figure 4.12 shows the approximation of the American geometric put option, calculated by learned and simulated expression, with respect to the number of inner trajectories for different maturities. Both expressions converge to the benchmark value for 2^9 outer trajectories and 2^{12} inner trajectories in 3 seconds.

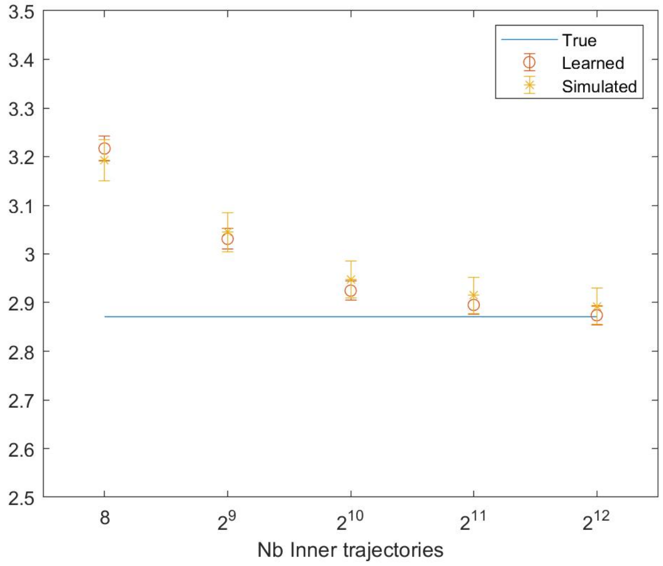


FIGURE 4.12 – Numerical approximation of price V_0 : $d_1 = 20$, $M_0 = 2^9$, $T = 1$, $L = 3$.

4.5.8 Initial Margin

Assume the setting presented in Section 4.3.2, we consider a portfolio of one hundred put options, the price V_{s_k} of the portfolio at time step s_k is given by

$$V_{s_k} = \sum_{i=0}^{d_1} e^{-(T-s_k)r} \mathbb{E}_{s_k}\Big(\big[K - X_T^i\big]^+\Big) \tag{4.5.8}$$

with an asset X given by $X_t^i = X_s^i \exp\Big((r - \frac{\sigma^2}{2})(t-s) + \sigma(W_t^i - W_s^i)\Big)$, $t > s$, $1 \leq i \leq d_1$, with r the interest rate, K the strike and T the maturity.

We are interested to calculate the initial margin (IM) of this portfolio. IM is an amount posted by the counterparty (or the bank) to overcome the loss of the portfolio during the liquidation period after a default.

IM is formalized here as follows

$$\mathrm{IM}_{s_k} = \mathbb{ES}_{s_k}^a(L_{s_k,s_k+\delta}) \tag{4.5.9}$$

where the loss of the portfolio at time t over a period δ is denoted $L_{s_k,s_k+\delta}$ and is defined here by

$$L_{s_k,s_k+\delta} = V_{s_k+\delta} - V_{s_k},$$

and the expected shortfall $\mathbb{ES}$ is defined by

$$\mathbb{ES}_{s_k}^a(X) = \frac{1}{(1-a)} \int_a^1 \mathbb{V}a\mathbb{R}_{s_k}^\alpha(X) d\alpha.$$

The value-at-risk of some random variable $\mathbb{V}a\mathbb{R}^\alpha(X)$ conditionally to $\mathcal{F}_{s_k}$ is defined by

$$\mathbb{V}a\mathbb{R}_{s_k}^\alpha(X) = \inf\{x \in \mathbb{R} : \mathbb{P}(X \leq x \mid \mathcal{F}_{s_k}) \geq \alpha\}.$$

We considered the following parameters : $T = 1$, $d_1 = 100$, $K = X_0^i = 100$, $r = 0.01$, $a = 99\%$, $NI = 32$ is the number of time step, $\delta = \frac{1}{32}$. A benchmark approximation of the IM is obtained using Black & Scholes formula for put options.

Figue 4.13 shows some distributions of the loss process. From top to bottom we show different time steps $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$. On the left, we perform the procedure without variance adjustment and on the right we perform the variance adjustment introduced in section 4.2.3. We show that the variance adjustment is necessary to fit the benchmark distribution of the loss process. Figure 4.14 shows the initial margin distribution. From top to bottom we show different time steps $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$. Although we are interested in distribution tails of the loss process we have a fairly good representation of the distribution of IM. Figure 4.15 shows at the top the mean of IM with respect to the time horizon of the portfolio and we show the L2 relative error at the bottom. The relative error is sufficiently small as it is generally less than 8% and does not exceed 11%.

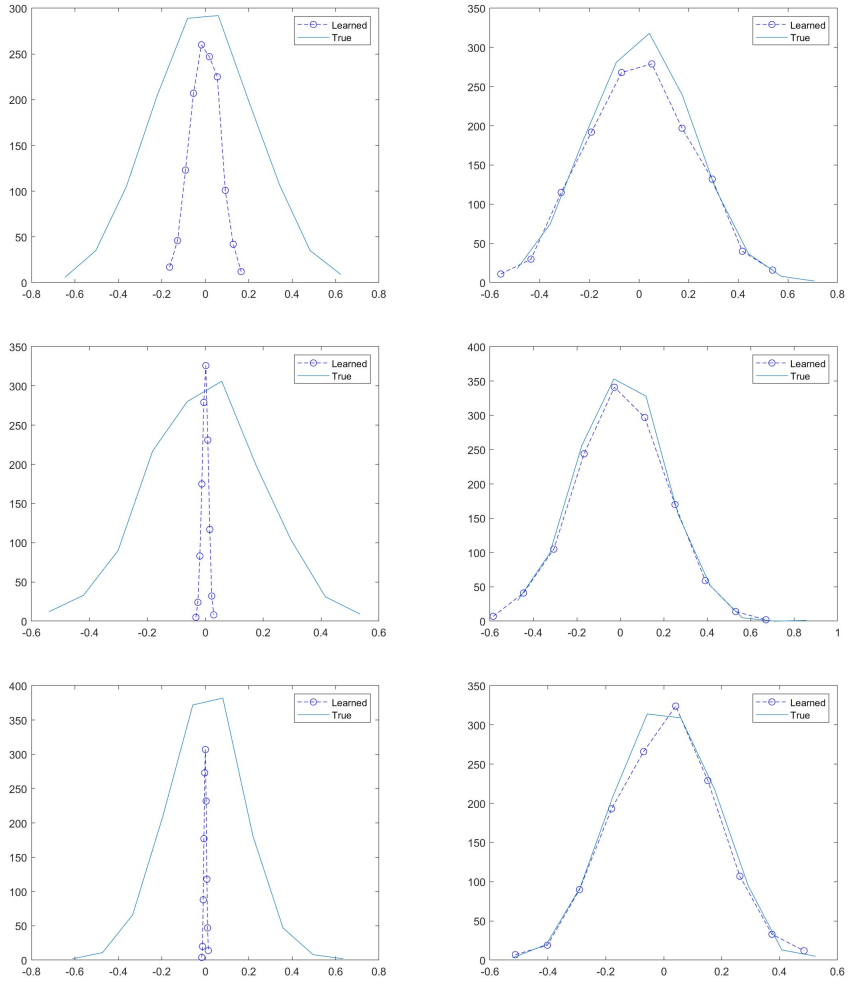


FIGURE 4.13 – Numerical approximation of the loss distribution [Left] Without variance adjustment, [Right] With variance adjustment ; [top to bottom] $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$; $M_0 = 2^8$, $M_1 = 2^8 * 5$.

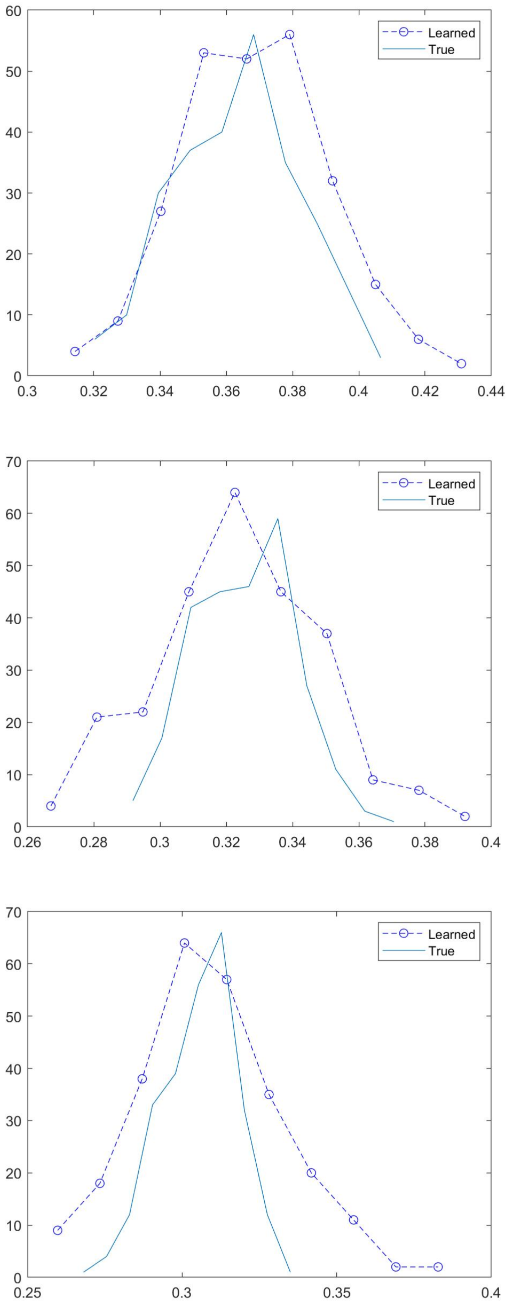


FIGURE 4.14 – Numerical approximation of the IM distribution : [top to bottom] $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$; $M_0 = 2^8$, $M_1 = 2^8 * 5$.

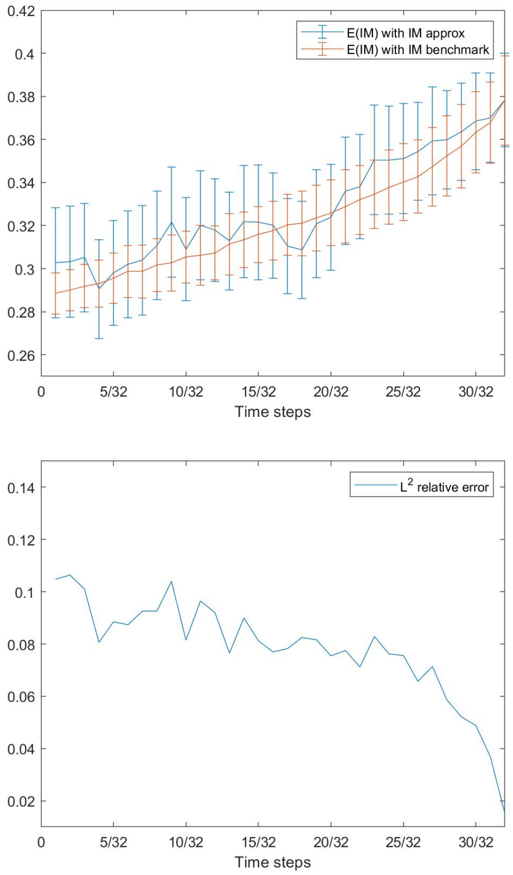


FIGURE 4.15 – Initial Margin : [Top] mean of IM_{s_k} ; [Bottom] L2 relative error.

Références

[1] L. Abbas-Turki, A.I. Bouselmi, and M.A. Mikou. Toward a coherent monte carlo simulation of cva. *Monte Carlo Methods and Applications*, 20(3) :195–216, 2014.

[2] L. Abbas-Turki, S Crépey, and B. Diallo. XVA principles, nested Monte Carlo strategies, and GPU optimizations. *International Journal of Theoritical and Applied Finance*, 2018.

[3] L. Abbas-Turki, B. Lapeyre, P. Mercier, and S. Vialle. High dimensional pricing of exotic european contracts on a gpu cluster, and comparison to a cpu cluster. 2009.

[4] L. Abbas-Turki, B. Lapeyre, P. Mercier, and S. Vialle. Pricing derivatives on graphics processing units using monte carlo simulation. *Concurrence and Computation : Pratice and Experience*, 26, 2014.

[5] L.A. Abbas-Turki and S. Graillat. Resolving small random symmetric linear systems on graphics processing units. *The Journal of Supercomputing*, 73(4) :1360–1386, 2017.

[6] A. Agarwal, S. De Marco, E. Gobet, and G. Liu. Study of new rare event simulation schemes and their application to extreme scenario generation. *Mathematics and Computers in Simulation*, (143) :89–98, 2018.

[7] C. Albanese and L. Andersen. Accounting for OTC derivatives : Funding adjustments and the re-hypothecation option. Working paper available at https ://papers.ssrn.com/sol3/papers.cfm ?abstract_id=2482955, 2014.

[8] C. Albanese and L. Andersen. FVA : What's wrong, and how to fix it. *Risk Magazine*, pages September 54–56, 2015.

[9] C. Albanese, L. Andersen, and S. Iabichino. FVA : Accounting and risk management. *Risk Magazine*, pages February 64–68, 2015. Long preprint version available as ssrn.2517301.

[10] C. Albanese, T. Bellaj, G. Gimonet, and G. oPietronero. Coherent global market simulations and securitization measures for counterparty credit risk. *Quantitative Finance*, 11(1) :1–20, 2011.

[11] C. Albanese, S. Caenazzo, and S. Crépey. Credit, funding, margin, and capital valuation adjustments for bilateral portfolios. *Probability, Uncertainty and Quantitative Risk*, 2(7) :26 pages, 2017. Preprint version available on https ://math.maths.univ-evry.fr/crepey.

[12] C. Albanese, S. Crépey, R. Hoskinson, and B. Saadeddine. XVA analysis from the balance sheet. *Working paper available at https ://math.maths.univ-evry.fr/crepey*, 2019.

[13] C. Albanese and H. Li. Monte Carlo pricing using operator methods and measure changes. Working paper available at https ://papers.ssrn.com/sol3/papers.cfm ?abstract_id=1484556, 2010.

[14] J. An and A. Owen. Quasi-regression. *Journal of Complexity*, pages 588–607, 2001.

[15] L. Andersen, D. Duffie, and Y. Song. Funding value adjustments. *Journal of Finance*, 74(1) :145–192, 2019.

[16] J. Andreasen. Cva on ipad mini. *Global Derivatives, Presentation*, 2014.

[17] J. Andreasen. Cva on ipad mini, part1. *Intro. Aarhus Kwant Factory PhD Course*, 2014.

[18] J. Andreasen. Cva on ipad mini, part1. *Cutting the it edge. Aarhus Kwant Factory PhD Course*, 2014.

[19] J. Andreasen. Cva on ipad mini, part2. *The beast. Aarhus Kwant Factory PhD Course*, 2014.

[20] J. Andreasen. Cva on ipad mini, part3. *Xva algorithms. Aarhus Kwant Factory PhD Course*, 2014.

[21] F. Anfuso, D. Aziz, K. Loukopoulos, and P. Giltina. A sound modelling and backtesting framework for forecasting initial margin requirements. *Risk Magazine*, 2017.

[22] M. Anthony and P.L. Bartlett. Neural network learning : Theoretical foundations. *Cambridge University Press*, 2009.

[23] J. Aquilina, G. Cesari, N. Charpillon, Z. Filipovic, I. Manda, and G. Lee. *Modelling, Pricing, and Hedging Counterparty Credit Exposure : A Technical Guide*. Springer, 2010.

[24] Y. Armenti and S. Crépey. Xva metrics for ccp optimization. *Working paper available at https ://math.maths.univ-evry.fr/crepey,*, 2018.

[25] J. Arnulf, H. Jiequn, and E. Weinan. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(1) :349–380, 2018.

[26] R. Avikainen. On irregular functionals of sdes and the euler scheme. *Finance and Stochastics*, 13 :381–401, 2009.

[27] A. Bally and G. Pagès. A quantization algorithm for solving discrete time multidimensional optimal stopping problems. *Bernoulli*, 6 :1003–1049, 2003.

[28] O. Bardou, N. Frikha, and G. Pagès. Recursive computation of value-at-risk and conditional value-at-risk using MC and QMC. *Monte Carlo and quasi-Monte Carlo methods*, 2008 :193–208, 2009.

[29] O. Bardou, N. Frikha, and G. Pagès. CVaR hedging using quantization-based stochastic approximation algorithm. *Math. Finance*, 26(1) :184–229, 2016.

[30] D. Barrera, S. Crépey, B. Diallo, G. Fort, E. Gobet, and U. Stazhynski. Stochastic approximation schemes for economic capital and risk margin computations. *ESAIM : PROCEEDINGS AND SURVEYS*, 65 :182–218, 2019.

[31] R. Bellma. Dynamic programming. *Princeton Landmarks in Mathematics, Princeton University Press*, 2010.

[32] A. Benveniste, M. Métivier, and P. Priouret. Adaptive algorithms and stochastic approximations. *Applications of Mathematics (New York)*, 22(1), 1990.

[33] M. Bichuch, A. Capponi, and S. Sturm. Arbitrage-free XVA. *Mathematical Finance*, 28(2) :582–620, 2018.

[34] T. R. Bielecki, D. Brigo, and S. Crépey. *Counterparty Risk and Funding : A Tale of Two Puzzles*. Chapman & Hall/CRC Financial Mathematics Series, 2014.

[35] B. Bouchard and N. Touzi. Discrete time approximation and monte carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Application*, 111 :175–206, 2004.

[36] D. Brigo and A. Pallavicini. Nonlinear consistent valuation of CCP cleared or CSA bilateral trades with initial margins under credit, funding and wrong-way risks. *Journal of Financial Engineering*, 1 :1–60, 2014.

[37] M. Broadie, Y. Du, and C. Moallemi. Risk estimation via regression. *Operations Research*, 63(5) :1077–1097, 2015.

[38] J. Bucklew. Introduction to rare event simulation. *Springer Series in Statistics*, 2004.

[39] G. Cesari, J. Aquilina, N. Charpillon, Z. Filipovic, G. Lee, and I. Manda. *Modelling, Pricing, and Hedging Counterparty Credit Exposure.* Springer Finance, 2010.

[40] J.-F. Chassagneux. Linear multistep schemes for bsdes. *SIAM J. Numer. Anal.*, 52(6) :2815–2836, 2014.

[41] J.-F. Chassagneux and A. Richou. Numerical simulation of quadratic bsdes. *Ann. Appl. Probab.*, 26(1) :262–304, 2016.

[42] E. Clément, D. Lamberton, and P. Protter. An analysis of least squares regression method for american option pricing. *Finance and Stochastics*, 6(4) :449–471, 2002.

[43] A. Coates, B. Huval, T. Wang, D. Wu, B.A Catanzaro, and N. Andrew. Deep learning with cots hpc systems. *Proceedings of the 30th International Conference on Machine Learning*, 28 :1337–1345, 2013.

[44] S. Crépey. *Financial Modeling : A Backward Stochastic Differential Equations Perspective.* Springer Finance Textbooks, 2013.

[45] S. Crépey, R. Élie, W. Sabbagh, and S. Song. When capital is a funding source : The xva anticipated bsdes. *Preprint version available at https ://math.maths.univ-evry.fr/crepey*, 2017.

[46] S. Crépey, G. Fort, E. Gobet, and U. Stazhynski. Uncertainty quantification for stochastic approxiamtion limits using chaos expension. 2017.

[47] S. Crépey and T.M. Nguyen. Nonlinear monte carlo schemes for counterparty risk on credit derivatives. *In Challenges in Derivatives Markets, Springer Proceedings in Mathematics*, pages 53–82, 2016.

[48] S. Crépey and A. Rahal. Simulation/regression pricing schemes for CVA computations on CDO tranches. *Communications in Statistics–Theory and Methods*, 43(7) :1390–1408, 2014.

[49] S. Crépey and S. Song. Counterparty risk and funding : Immersion and beyond. *Finance and Stochastics*, 20(4) :901–930, 2016.

[50] J.-F. Delmas and B. Jourdain. Modèles aléatoires. volume 57 of *Mathématiques et Applications*. Springer, 2006.

[51] M. Duflo. *Algorithmes stochastiques.* Springer, Berlin, 1996.

[52] Y. Elouerkhaoui. Pricing and hedging in a dynamic credit model. *International Journal of Theoretical and Applied Finance*, 10(4) :703–731, 2007.

[53] Y. Elouerkhaoui. From FVA to KVA : including cost of capital in derivatives pricing. *Risk Magazine*, pages March 70–75, 2016.

[54] Y. Elouerkhaoui. The funding invariariance principle. *Risk*, 2016.

[55] Y. Elouerkhaoui. Credit correlation : Theory and practice. *Palgrave Macmillan*, 2017.

[56] P. Embrechts, C. Klueppelberg, and T. Mikosch. Modelling extremal events for insurance and finance. volume 33 of *Applications of Mathematics*. Springer, 1997.

[57] G. Fort. Central limit theorems for stochastic approximation with controlled Markov chain dynamics. *ESAIM Probab. Stat*, 19 :60–80, 2010.

[58] M. Fujii and A. Takahaski. Analytical approximation for non-linear fbsdes with perturbation scheme. *International Journal of Theoretical and Applied Finance*, 15(5), 2012.

[59] M. Fujii and A. Takahaski. Perturbative expansion of fbsde in an incomplete market with stochastic volatility. *Quarterly Journal of Finance*, 2(3), 2012.

[60] S. Gerhold. The Longstaff-Schwartz algorithm for Lévy models : Results on fast and slow convergence. *The Annals of Applied Probability*, 21(2) :589–608, 2011.

[61] M. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56 :607–617, 2005.

[62] P. Glasserman. *Monte Carlo methos in finance engineering.* Stochastics Modelling and Applied Probability, Springer-Verlag, New York, 2003.

[63] P. W. Glynn and S. H. Lee. Computing the distribution function of conditional expectation via monte carlo : Discrete conditioning spaces. *ACM Transactions on Modeling and Computer Simulation*, 13(5) :238–258, 2003.

[64] E. Gobet and C. Labart. Error expansion for the discretization of backward stocahstic differential equations. *Stochastic Processes and their Applications*, pages 803–829, 2007.

[65] E. Gobet, J.-P. Lemor, and X. Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3) :2172–2202, 2005.

[66] E. Gobet and S. Pagliarani. Analytical approximations of bsdes with nonsmooth driver. *SIAM Journal on Financial Mathematics*, 6 :919–958, 2015.

[67] E. Gobet and P. Turkedjiev. Linear regression mdp scheme for discrete backward stochastic differential equations under general conditions. *Mathematics of Computation*, (85) :1359–1391, 2016.

[68] M.B. Gordy and S. Juneja. Nested simulation in portfolio risk measurement. *Management Science*, 56(10) :1833–1848, 2010.

[69] A. Green. *XVA : Credit, Funding and Capital Valuation Adjustments.* Wiley, 2015.

[70] A. Green and C. Kenyon. Efficient xva management : Pricing, hedging, and attribution using trade-level regression and global conditioning. https ://arxiv.org/abs/1412.5332, 2014.

[71] A. Green and C. Kenyon. MVA by replication and regression. *Risk Magazine*, pages May 82–87, 2015.

[72] A. Green, C. Kenyon, and C. Dennis. KVA : capital valuation adjustment by replication. *Risk Magazine*, pages December 82–87, 2014. Preprint version “KVA : capital valuation adjustment” available at ssrn.2400324.

[73] O. Green, R. Mccool, and D. Bader. Gpu merge path : A gpu merging algorithm. *In 26th ACM International Conference on Supercomputing (ICS)*, pages 25–29, 2012.

[74] P. Hall and C. C. Heyde. *Martingale limit theory and its application.* Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York, 1980.

[75] P. Henry-Labordere. Counterparty risk valuation : A marked branching diffusion approach. *The review of Financial Studies*, 2012.

[76] B. Huge and A Savinne. Lsm reloaded, differentiate xva on your ipad mini. Working paper available at SSRN : https ://ssrn.com/abstract=2966155, 2017.

[77] J. Hule and A White. The fva debate. *Risk*, pages 83–85, 2012.

[78] J. Hull and A. White. CVA and wrong way risk. *Financial Analyst Journal*, 68 :58–69, 2012.

[79] M. Iben Taarit. *Pricing of XVA Adjustments : from Expected Exposures to Wrong-Way risks.* PhD thesis, Université Paris-Est, 2017. Forthcoming.

[80] B. Jourdain and J. Lelong. Robust adaptative importance sampling for normal random vectors. *The Annals of Applied Probability*, 19(5) :1687–1718, 2009.

[81] N. El Karoui, S. Peng, and C. M. Quenez. Backward stochastics differential equations in finance. *Mathematical Finance*, 7(1) :349– 380, 1997.

[82] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statis.*, (23), 1990.

[83] H. J. Kushner and G. G. Yin. Stochastic approximation and recursive algorithms and applications. *Applications of Mathematics (New York)*, 35(2), 2003.

[84] A. Krzyżak L. Györfi, M. Kohler and H. Walk. *A distribution-free theory of nonparametric regression.* Springer Series in Statistics, New York, 2008.

[85] P. L'Ecuyer. Combined multiple recursive random number generators. *Operations Research*, 44(5), 1996.

[86] S.H. Lee. Monte carlo computation of conditional expectation quantiles. *Ph.D thesis, Stanford University*, 1998.

[87] P. J. Lemor, E. Gobet, and X. Warin. Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli*, 12(5) :889–916, 2006.

[88] Minqiang Li and Fabio Mercurio. Jumping with default : wrong-way risk modelling for CVA. *Risk Magazine*, page November, 2015.

[89] J.-L. Lions, Y. Maday, and G. Turinici. A parareal in time discretization of pde's. *Comptes rendus de l'Académie des Sciences*, 1(1) :661–668, 2015.

[90] F. A. Longstaff and E. S. Schwartz. Valuing american options by simulation : A simple least square approach. *The review of Financial Studies*, 14(1) :113–147, 2001.

[91] Lee Moran and Sascha Wilkens. Capturing initial margin in counterparty risk calculations. *Journal of Risk Management in Financial Institutions*, 10 :118–129, 2017.

[92] W. K. Newey. Convergence rates and asymptotic normality for series estimators. *Journal of Econometrics*, 79 :147–168, 1995.

[93] Nvidia. Cuda C best practices guide. 2017.

[94] Nvidia. Cuda C programming guide. 2017.

[95] Nvidia. Nvidia NVLink TM high-speed interconnect : Application performance. 2017.

[96] Committee of European Insurance and Occupational Pensions Supervisors. QIS5 technical specifications. 2010.

[97] Swiss Federal Office of Private Insurance. Technical document on the Swiss solvency test. pages 21–42, 2006.

[98] G. Pagès. Numerical probability : an introduction with applications to finance. *Universitext, Springer*, 2002.

[99] G. Pagès and A. Sagna. Improved error bounds for quantization based numerical schemes for bsde and nonlinear filtering. *Stochastic Processes and their Applications*, 128 :847–883, 2018.

[100] E. Pardoux and S. Peng. Adapted solution of a backward stochastics differential equation. *Systems and Control Letters*, 14 :55–51, 1990.

[101] B. T. Polyak. New stochastic approximation type procedures. *Automation and remote control*, 51 :98–107, 1990.

[102] B.T. Polyak and A.B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control and Optimization*, 30 :838–855, 1992.

[103] W. H. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C++, The Art of Scientific Computing.* Cambridge University Press, New York, 2007.

[104] M. Pykhtin. General wrong-way risk and stress calibration of exposure. *Journal of Risk Management in Financial Institution*, 5 :234–251, 2012.

[105] T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On the opportunities and pitfalls of nesting monte carlo estimators. arXiv :1709.06181v3, 2017.

[106] A. Reghai, O. Kettani, and M. Messaoud. Cva with greeks and aad. *Risk Magazine, December*, 2015.

[107] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat*, 22 :400–407, 1951.

[108] H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. *Optimizing methods in statistics (Proc. Sympos., Ohio State Univ., Columbus*, pages 233–257, 1971.

[109] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2 :493–517, 2000.

[110] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26 :1443–1471, 2002.

[111] S.-Li and N. Amenta. Brute-force k-nearest neighbors search on the gpu. *CIn Similarity Search and Applications – Springer*, pages 259–270, 2015.

[112] D P. Singh, I. Joshi, and J. Choudhary. Survey of gpu based sorting algorithms. *International Journal of Parallel Programming*, 46 :1017–1034, 2018.

[113] J. N. Tsitsiklis and B. Van Roy. Regression methods for pricing complex american-style options. *IEEE Transactions on neural networks*, 12(4), 2001.

[114] H. White. Asymtotic theory for econometricians. *Academic Press*, 2001.

universite
PARIS-SACLAY

École doctorale
de mathématiques
Hadamard (EDMH)

Titre : Calcul de XVAs par simulation imbriquée sur Graphics Processing Units

Mots Clefs : XVA, Simulation imbriquée, Graphics Processing Units

Résumé : Cette thèse traite du calcul des X-valorisations d'ajustement, où X englobe C pour le crédit, F pour le financement, M pour la marge et K pour le capital. Nous étudions différentes approches basées sur la simulation imbriquée et implémentées sur des unités de traitement graphique (GPU). Nous examinons d'abord le problème, pour une banque ou une assurance, du calcul numérique de son capital économique sous la forme d'une "value-at-risk" ou d'une "expected shortfall" de sa perte sur un horizon de temps donné. En utilisant une approche d'approximation stochastique pour ces mesures de risque, nous établissons la convergence des schémas résultant de la simulation du capital économique. Ensuite, nous présentons une approche de Monte Carlo imbriqué (NMC) pour le calcul des XVAs. Nous montrons que le calcul global des XVAs implique cinq niveaux de dépendance. Les couches les plus hautes sont d'abord lancées et déclenchent des simulations imbriquées à la volée si nécessaire pour calculer un élément à partir d'une couche inférieure. Enfin, nous présentons un algorithme basé sur un Monte Carlo imbriqué à une couche (1NMC) pour simuler les fonctions U d'un processus de Markov X. La principale originalité de la méthode proposée provient du fait qu'elle fournit une recette pour simuler $U_{t\geq s}$ conditionnellement à X_s. La généralité, la stabilité et le caractère itératif de cet algorithme, même en haute dimension, en font la force.

Title : X-Valuation Adjustment Computations by Nested Simulation on Graphics Processing Units

Keys words : XVA, Nested simulation, Graphics Processing Units

Abstract : This thesis deals with X-valuation adjustment computation, where X ranges over C for credit, F for funding, M for margin, and K for capital. We investigate different approaches based on nested simulation and implemented on graphics processing units (GPU). First, we consider the problem of the numerical computation of its economic capital by a bank or an insurance, in the form of a value-at-risk or expected shortfall of its loss over a given time horizon. Using a stochastic approximation point of view on these risk measures, we establish the convergence of the resulting economic capital simulation schemes. Then we present a nested Monte Carlo (NMC) approach for XVA computations. We show that the overal XVA suite involves five compounded layers of dependence. Higher layers are launched first and trigger nested simulations on-the-fly whenever required in order to compute an item from a lower layer. Finally, we present an algorithm based on a one-layered Nested Monte Carlo (1NMC) to simulate functionals U of a Markov process X. The main originality of the proposed method comes from the fact that it provides a recipe to simulate $U_{t\geq s}$ conditionally on X_s. The generality, the stability and the iterative nature of this algorithm, even in high dimension, make its strength.